# BILINEAR AUTOENCODERS FIND INTERPRETABLE MANIFOLDS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Sparse autoencoders have become a standard tool for uncovering interpretable latent representations in neural networks. Yet salient concepts often span manifolds that current linear methods cannot capture without post hoc analysis. This paper uses quadratic latents to close this gap using bilinear autoencoders, which decompose activations into low-rank quadratic forms, compose linearly in weight space, and admit input-independent geometric analysis. This qualitative difference in what concepts quadratic latents can detect challenges the standard linear representation hypothesis. Our experiments and visualisations show that multi-dimensional geometries are highly prevalent in language models and that composite latents capture them at a reconstruction quality and interpretability competitive with sparse autoencoders. Furthermore, we show that autoencoders with varying geometric priors recover the same input subspace despite their dictionary entries being distinct. Practically, these models serve as an unsupervised tool for manifold discovery, which we demonstrate through an interactive online visualizer for Qwen 3.5. This is a step toward nonlinear but mathematically tractable latent representations whose composition is expressive and interpretable by design.

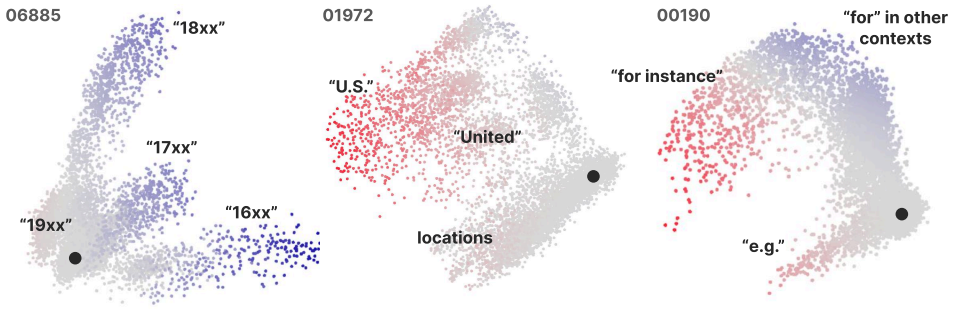


Figure 1: Three hand-picked latent features from our interactive viewer using Qwen 3.5: `https://bae-9xf.pages.dev`. Our proposed bilinear autoencoders capture low-dimensional manifolds. The above visuals are created by linearly projecting activations of input samples into the (3D) eigenspace of three different dictionary elements. Point colour shows the signed activation of that latent (blue is positive, red is negative) and the black dot marks the origin of the activation space. The left latent fires selectively on years whose geometry represents them as superposed linear subspaces. The middle latent activates on U.S.-related tokens, and the inactive (grey) samples correspond to other locations. The geometry is scattered across all 3 dimensions, containing visible semantic clusters. The right latent activates positively on 'for instance' and 'for example' and negatively on 'for' in other contexts. Refer to subsection 4.2 for further explanation and to Appendix K for random examples.

## 1 INTRODUCTION

Neural networks are highly nonlinear systems shaped by emergent rather than explicit mechanisms. Logical human understanding, by contrast, is grounded in explicit structure and predictable composition. Mechanistic interpretability attempts to bridge this gap through reverse-engineering, identifying interpretable bases (Olah, 2022; Cunningham et al., 2023) and the circuits that act upon them. This lets us examine latent features in isolation, despite their co-dependence (Ameisen et al., 2025). Yet



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

the core problem remains: these latents often lack a compositional interpretation, since their isolated behaviour does not determine their role in the whole (Gauderis et al., 2026; Coecke, 2021). Extracted latents have implicit domains of applicability, regions in input or activation space within which they are meaningful. Latent splitting and absorption (Chanin et al., 2025) illustrate this. A latent's meaning is not fixed but depends on which other latents are active alongside it, so isolated study leaves this context-dependence invisible (Bhalla et al., 2026). The isolated study of latents is therefore of limited value, because their interaction may skew interpretation even at small scales (Méloux et al., 2025). Compositional understanding requires interactions that are explicit and computable (Tull et al., 2024; Gauderis et al., 2026).

This paper develops bilinear autoencoders as a lens on representation geometry (Figure 1). They decompose activations into a dictionary of quadratic latents whose receptive fields admit closed-form geometric analysis. We motivate the architecture with a framework of structural priors that map directly to manifold shapes, and validate them against modern language models. We make three contributions[1]

- **Architecture.** We adapt bilinear autoencoders to dictionary learning, with efficient training via a kernel trick, and motivate the architecture as a natural and tractable extension of existing linear sparse autoencoders (section 2).
- **Framework.** We propose three nested representation hypotheses (atomic, composite, quadratic), formalising how structural priors on the dictionary translate into receptive-field geometries: slabs, ellipsoids and hyperboloids, and arbitrary quadrics (section 3).
- **Empirics.** Multi-dimensional geometries are prevalent in language models and composite latents capture them with lower reconstruction error while remaining interpretable. Bilinear autoencoders recover stable input subspaces across priors despite learning disjoint dictionaries (section 4).

We write scalars as $s$, vectors as $\mathbf{v}$, matrices as $M$ and higher-order tensors as $\mathbf{T}$, and use colour to distinguish **weight-based** from **activation-based** quantities (Appendix A).

## 2 Bilinear autoencoders

Autoencoders reconstruct their inputs through a bottleneck, forcing the model to capture structure in the data (Hinton & Salakhutdinov, 2006; Hindupur et al., 2025). They have become a key tool in interpretability for extracting meaningful latents from neural representations (Olshausen & Field, 1996; Cunningham et al., 2023; Bricken et al., 2023). Their interpretative goal is dual: reconstruct the input, and trace the extracted latents back to the original model. These goals pull against each other, trading expressivity for analysability.

The truncated singular value decomposition is itself a dimensionality-reducing autoencoder, but its compression is too restrictive: neural representations typically span all dimensions, so rank-based methods miss much of their structure (Elhage et al., 2022). Sparse autoencoders instead focus on sparsity in a high-dimensional latent space, adding a sparsity-inducing nonlinearity to the encoder while keeping the decoder linear (Lewicki & Sejnowski, 2000; Aharon et al., 2006; Mairal et al., 2009). They recover linearly separable latents that trace back to the original representations. Because each approach assumes a fixed geometry, the resulting latents mirror the imposed prior instead of the underlying structure of the data. (Engels et al., 2025; Buchanan et al., 2025; Hindupur et al., 2025).

### 2.1 Geometric priors for dictionary learning

Current autoencoder architectures rest on the *linear representation hypothesis*: the claim that neural representations encode concepts mostly along linear directions (Park et al., 2024; Cunningham et al., 2023; Gao et al., 2024; Rajamanoharan et al., 2024). Yet, recent research shows that salient concepts often span more intricate manifolds that existing methods cannot capture (Modell et al., 2025; Engels et al., 2025; Bhalla et al., 2026; Fel et al., 2026). Increasing capacity is a natural response, but unbounded expressivity collapses to trivial solutions and defeats interpretation (Gao et al., 2024; Sutter et al., 2025).

[1]The code is publicly available at [see supplementary material].

Using autoencoders for interpretability is inherently about hypotheses (Hindupur et al., 2025) and finding the lens under which representations become understandable. Yet we do not know the neural representation geometry in advance; autoencoder architecture and its training should guide discovery toward simple latents without prescribing their form. We need primitives that are simple individually but compose into latents of varying complexity (Jenatton et al., 2011; Bhalla et al., 2026). This requires measuring the complexity of the learned feature class itself, not just of the activations it produces. Minimum description length (Gauderis et al., 2026; Rissanen, 1978; Grünwald, 2007; Ayonrinde et al., 2024) is the natural candidate, but it requires a model class in which architectural description length independent of data is well-defined. Polynomials fit this role: their closed-form composition produces increasingly intricate, yet well-understood manifolds in their input space. Their parameters combine linearly, so latents can be added, decomposed or compared in weight space, with no forward pass and no data (Dooms et al., 2024; Pearce et al., 2025; Lewis, 2019). Recent work instead documents representation geometry post-hoc, from activation statistics and often on top of a trained sparse autoencoder (Engels et al., 2025; Modell et al., 2025; Bhalla et al., 2026), which recovers the empirical geometry but severs its tie to the parameters that produced it.

### 2.2 From gated units to quadratic latents

The simplest nonlinear instance of this idea is to model latents as quadratics. We implement this using bilinear autoencoders, which adapt sparse autoencoders to gated linear units (Shazeer, 2020; Dauphin et al., 2017). These units apply two matrices $\boldsymbol{L}, \boldsymbol{R} \in \mathbb{R}^{h \times d}$ to their input $\mathbf{x} \in \mathbb{R}^d$ and project their element-wise multiplication using $\boldsymbol{C} \in \mathbb{R}^{k \times h}$.

$$\mathrm{GLU}(\mathbf{x}) \;=\; \boldsymbol{C}\big(\sigma(\boldsymbol{L}\mathbf{x}) \odot \boldsymbol{R}\mathbf{x}\big)$$

Removing the gating activation $\sigma$ has a negligible impact on accuracy (Pearce et al., 2025; Shazeer, 2020). What remains computes second-order polynomials of $\mathbf{x}$, but remains linear in the 'lifted' *product space* of pairwise interactions $\boldsymbol{X} = \mathbf{x}\mathbf{x}^\top$. The $i$-th latent activation is

$$\boldsymbol{z}_i \;:=\; \mathbf{x}^\top \boldsymbol{L}^\top \operatorname{diag}(\mathbf{c}_i)\, \boldsymbol{R}\mathbf{x} \;=\; \langle \boldsymbol{W}_i,\, \boldsymbol{X} \rangle_F$$

where $\mathbf{c}_i$ is the $i$-th row of $\boldsymbol{C}$ and $\boldsymbol{W}_i$ is the bilinear form of the $i$-th latent,

$$\boldsymbol{W}_i \;=\; \sum_{j \in \mathrm{supp}(\mathbf{c}_i)} \boldsymbol{C}_{ij}\, \mathbf{l}_j \mathbf{r}_j^\top .$$

Stacking all $\boldsymbol{W}_i$ into a third-order tensor gives the full encoder $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{k \times (d \times d)}$. The autoencoder's training objective is to approximate $\boldsymbol{X}$ through a symmetric bottleneck:

$$\hat{\boldsymbol{X}} \;=\; \boldsymbol{\mathcal{W}}^\top \boldsymbol{\mathcal{W}}\, \boldsymbol{X}.$$

Lifting the input to $\boldsymbol{X} = \mathbf{x}\mathbf{x}^\top$ turns the autoencoding problem into a linear factorisation of quadratic latents. This representation makes the geometric structure explicit: quadric activation regions, such as ellipsoids and hyperboloids, are level sets of linear functionals of $\boldsymbol{X}$ (section 3; Figure 3).[2] This lifted space has $\Theta(d^2)$ dimensions, making direct materialisation of $\boldsymbol{X}$ intractable at scale; we instead evaluate the loss via a kernel trick (Appendix D), which scales quadratically in compute with the latent count $k$ while its memory use matches an ordinary autoencoder. Since the model cannot degenerate to identity-like maps—as ordinary sparse autoencoders do without sparsity penalties—its bottleneck lies in which combinations to preserve. In other words, the learned $\boldsymbol{\mathcal{W}}$ specifies exactly which second-order structure is kept and discarded.

### 2.3 Scale-invariant activation sparsity

The choice of $\boldsymbol{\mathcal{W}}$ governs which second-order patterns survive; activation sparsity governs how cleanly each fires. The two are complementary. The standard approach minimises sparsity-inducing norms ($L_{0<p\leq 1}$) on activations, but causes trivial shrinkage and can destabilise training by producing dead latents (Bricken et al., 2023). Both pull against our goal: we want *relative* prominence across

[2]The lifted form is homogeneous, so these quadrics are centred at the origin. Augmenting the input before lifting, $\tilde{\mathbf{x}} = [\mathbf{x}; , 1]$, adds the terms that would displace them, but leaves our results unchanged. We suspect the model simply learns to repurpose a direction that stays approximately constant as a bias.

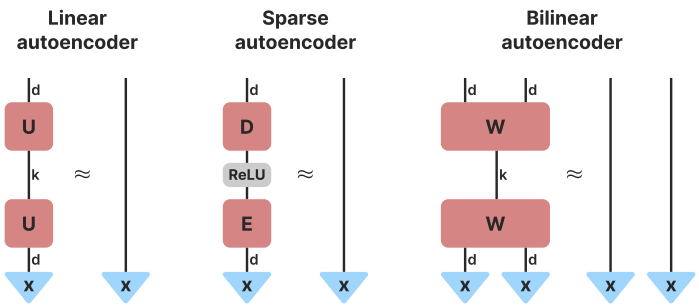


Figure 2: String diagrams (Coecke & Kissinger, 2018) of three autoencoders, read bottom-up. Boxes represent operations and wires the spaces they act on. Most autoencoders reconstruct their input nonlinearly. The bilinear encoder instead has two input wires, linearly reconstructing $\boldsymbol{X} = \mathbf{x}\mathbf{x}^\top \cong \mathbf{x} \otimes \mathbf{x}$ in the $d \times d$ product space, which is quadratic in $\mathbf{x}$.

the sample dimension $\mathbf{z}_i \in \mathbb{R}^n$ (effective batch size $n =$ samples $\times$ sequence length), not uniformly small activations (Hoyer, 2004).

$$\text{Hoyer}_{\text{density}}(\mathbf{z}_i - \boldsymbol{b}_i) \;=\; \frac{\|\mathbf{z}_i - \boldsymbol{b}_i\|_1 / \|\mathbf{z}_i - \boldsymbol{b}_i\|_2 - 1}{\sqrt{n} - 1} \tag{1}$$

This measure runs from 0 for one-hot vectors to 1 for perfectly uniform ones, and is invariant to scale by construction and, through $\boldsymbol{b}_i$, to translation: each latent learns the offset that makes its activation distribution as sparse as possible. The penalty is minimised when activations concentrate around any level set $\mathbf{z}_i = \boldsymbol{b}_i$, rewarding latents whose level sets align with structure in the data. It pressures latents to fire selectively on inputs without collapsing to zero (Bussmann et al., 2024). The training objective averages this penalty and adds it to the reconstruction error, weighted by $\alpha = 0.3$ (Appendix B).

$$\mathcal{L}(\hat{\boldsymbol{X}}, \boldsymbol{X}) \;=\; \frac{\|\hat{\boldsymbol{X}} - \boldsymbol{X}\|_F^2}{\|\boldsymbol{X}\|_F^2} \;+\; \frac{\alpha}{k} \sum_{i=1}^{k} \text{Hoyer}_{\text{density}}(\mathbf{z}_i - \boldsymbol{b}_i)$$

## 3 Structural Constraints of Bilinear Autoencoders

Any sparse autoencoder enforces two kinds of structural constraints: the decoder operationalises a hypothesis about how neural representations are constructed from concepts (Fel et al., 2025), while the encoder's receptive fields encode assumptions about how those concepts appear in the activation space (Hindupur et al., 2025). In bilinear autoencoders, the encoder and decoder are mirrors of each other. This section unpacks what each side implies, working through three nested hypotheses of increasing structure: the quadratic hypothesis (subsection 3.1), the atomic refinement (subsection 3.2), and the composite refinement (subsection 3.3).

### 3.1 The Quadratic Representation Hypothesis

Sparse autoencoders operationalise the *linear representation hypothesis* (Elhage et al., 2022; Fel et al., 2025). Representations decompose into a sparse sum of directions.

**Definition 1** (Linear Representation Hypothesis)**.** A distribution over representations $\mathbf{x} \in \mathbb{R}^d$ satisfies the *linear representation hypothesis* if there exists a single dictionary of directions $\{\mathbf{w}_i\}_{i=1}^k \subset \mathbb{R}^d$, such that every $\mathbf{x}$ in its support admits a sparse activation vector $\mathbf{z} \in \mathbb{R}^k$ satisfying $\mathbf{x} = \sum_{i \in \text{supp}(\mathbf{z})} \boldsymbol{z}_i \, \mathbf{w}_i$.

Bilinear decoders instead reconstruct $\boldsymbol{X} = \mathbf{x}\mathbf{x}^\top$, which lifts the problem to linear dictionary learning on the *Veronese variety* $\mathcal{V} = \{\mathbf{x}\mathbf{x}^\top : \mathbf{x} \in \mathbb{R}^d\} \subset \mathbb{R}^{d \times d}_{\text{sym}}$. This is the simplest non-linear relaxation of the linear hypothesis.

**Definition 2** (Quadratic Representation Hypothesis)**.** A distribution over representations $\mathbf{x} \in \mathbb{R}^d$ satisfies the *quadratic representation hypothesis* if there exists a single dictionary of symmetric matrices $\{\boldsymbol{W}_i\}_{i=1}^k \subset \mathbb{R}^{d \times d}_{\text{sym}}$, such that every $\mathbf{x}$ in its support admits a sparse activation vector $\mathbf{z} \in \mathbb{R}^k$ satisfying $\mathbf{x}\mathbf{x}^\top = \sum_{i \in \text{supp}(\mathbf{z})} \boldsymbol{z}_i \, \boldsymbol{W}_i$.

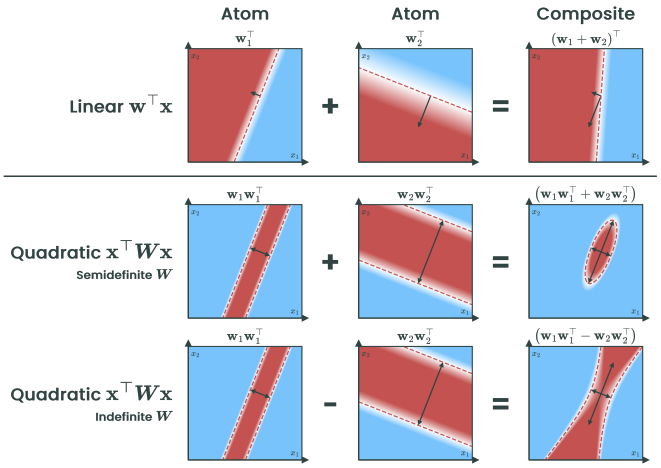


Figure 3: **Linear atoms gain no expressivity under composition; quadratic atoms compose algebraically to describe quadric geometry.** (*Top*) Linear atoms are signed half-space detectors, and their sum is another half-space which averages their directions. (*Middle and bottom*) Quadratic atoms are symmetric slabs that measure energy along a direction but ignore phase. Composing two rank-1 forms yields a rank-2 symmetric matrix, which unlocks quadric geometries. Their sum is a semidefinite *ellipsoid*, a conjunction detector for $\mathbf{w}_1$ *and* $\mathbf{w}_2$; their difference is a *hyperboloid*, a detector for $\mathbf{w}_1$ *and not* $\mathbf{w}_2$.

We adopt this hypothesis as the operationalising assumption required for any form of dictionary learning (Nguyen et al., 2012; Fel et al., 2025). Whether neural representations actually satisfy the quadratic hypothesis is an empirical question which we address in section 4. Under it, the dictionary atoms $\boldsymbol{W}_i$ are symmetric matrices instead of linear directions. Each matrix encodes a specific pattern of neurons that co-vary together, with diagonal entries $\boldsymbol{x}_i^2$ capturing the energy of individual dimensions and off-diagonal entries $\boldsymbol{x}_i\boldsymbol{x}_j$ encoding pairwise co-activations. Reconstructing $\boldsymbol{X}$ sparsely is therefore equivalent to decomposing the sample's covariance structure into a small set of fundamental interaction patterns, equivalent to reconstruction ICA on the Veronese embedding (Le et al., 2011). Instead of asking which directions activate, bilinear autoencoders ask *which co-occurrence patterns are active*.

Two consequences follow from reconstructing $\boldsymbol{X}$ instead of $\mathbf{x}$. First, a *second-order constraint*: a feature that survives training must describe how dimensions activate together, which is a harder constraint than what individual dimensions are active. Second, an *antipodal symmetry*: since $(-\mathbf{x})(-\mathbf{x})^\top = \mathbf{x}\mathbf{x}^\top$, the model ignores global phase, making it a good choice to detect features stored in antipodal superposition (Elhage et al., 2022) where a sparse autoencoder would have to spend two features on what is geometrically one direction.

### 3.2 Rank-1 atoms and energy separability

Just as sparse autoencoders build their dictionary from vectors, the simplest linear objects, bilinear autoencoders are naturally studied when their atoms are rank-1 matrices, the simplest quadratic objects.

**Definition 3** (Atomic Representation Hypothesis)**.** A distribution satisfies the *atomic representation hypothesis* if the dictionary atoms in the quadratic representation hypothesis are rank-1: $\boldsymbol{W}_i = \mathbf{w}_i\mathbf{w}_i^\top$.

Reconstructing $\mathbf{x}\mathbf{x}^\top = \sum_{i\in\mathrm{supp}(\mathbf{z})} \boldsymbol{z}_i\, \mathbf{w}_i\mathbf{w}_i^\top$ assumes that the covariance structure decomposes into a sparse sum of rank-1 prototypes; a few directions alone jointly account for the observed interactions.

**Receptive fields.** The encoder traces this hypothesis back to the input space and determines what concept geometry can be detected. A sparse autoencoder computes a signed projection and thresholds it, but an atomic bilinear encoder computes a squared projection:

$$\boldsymbol{z}_i^{\mathrm{lin}} = \mathrm{ReLU}(\mathbf{w}_i^\top\mathbf{x}), \qquad \boldsymbol{z}_i^{\mathrm{atom}} = (\mathbf{w}_i^\top\mathbf{x})^2 = \langle \mathbf{w}_i\mathbf{w}_i^\top, \boldsymbol{X}\rangle_F. \tag{2}$$

The second equality shows that each bilinear measurement is a linear functional on $\boldsymbol{X}$, the inner product of the Veronese image with the rank-1 atom $\mathbf{w}_i\mathbf{w}_i^\top$ which is equivalent to a Born-rule measurement (Coecke & Kissinger, 2018). These measurements form a projection encoder (Hindupur et al., 2025) from the Veronese variety onto the space of diagonal matrices.

The two receptive fields, for some threshold $\tau > 0$, are different:

$$\mathcal{F}_i^{\text{lin}} = \{\mathbf{x} : \mathbf{w}_i^\top \mathbf{x} > \tau\}, \qquad \mathcal{F}_i^{\text{atom}} = \{\mathbf{x} : (\mathbf{w}_i^\top \mathbf{x})^2 > \tau\}. \tag{3}$$

The sparse autoencoder's is a directed half-space (Hindupur et al., 2025); the bilinear's is a *symmetric slab*, two parallel hyperplanes around $\mathbf{w}_i^\perp$ and symmetric about the origin. Where a linear atom measures *which side of a hyperplane* $\mathbf{x}$ *lies on*, a quadratic atom measures *how close to a hyperplane* $\mathbf{x}$ *is*, regardless of sign. The respective data assumptions are *linear separability* (a concept is detectable from the sign and magnitude of a projection) versus *energy separability* (a concept is detectable from its magnitude alone). See Figure 3 for a visual comparison[3].

**Intrinsic sparsity.** The slab geometry of quadratic atoms has the advantage that they are more selective than linear ones in high dimensions. The proof is in Appendix E.

**Theorem 1** (Exponential receptive field gap)**.** *Let* $\mathbf{w} \in S^{d-1}$ *and* $\mathbf{x} \sim \text{Unif}(S^{d-1})$. *For any fixed threshold* $\tau \in (0, 1)$,

$$\frac{\mathbb{P}\big((\mathbf{w}^\top \mathbf{x})^2 > \tau\big)}{\mathbb{P}\big(\mathbf{w}^\top \mathbf{x} > \tau\big)} \;=\; \Theta\Big(e^{-d\tau(1-\tau)/2}\Big).$$

The receptive fields of quadratic atoms shrink exponentially faster with dimension than those of linear atoms. At $\tau = 1/2$ and $d = 1024$, the ratio is $\approx 10^{-56}$. A quadratic latent is therefore less likely to activate strongly without an additional constraint such as TopK or BatchTopK (Gao et al., 2024; Bussmann et al., 2024) that couple each receptive field to the competing activations.

### 3.3 Composite atoms and quadric separability

Because summing two sparse-autoencoder atoms yields another reoriented half-space, linear atoms compose poorly (Figure 3). Bhalla et al. (2026) identify this as the reason sparse autoencoders tile manifolds through many fragmented atoms rather than capturing them with one compact group, and call for featurisers that treat geometric objects as primitives. Quadratic atoms offer a solution. When two two rank-1 forms are composed, they produce a rank-2 symmetric matrix whose receptive field is an ellipsoid or hyperboloid instead of a slab (Figure 3). We formalise the hypothesis that concepts exploit this structure.

**Definition 4** (Composite Representation Hypothesis)**.** A distribution satisfies the *composite representation hypothesis* if the dictionary atoms in the quadratic representation hypothesis are sparse mixtures of rank-1 matrices over a shared base dictionary $\{\mathbf{w}_j\}_{j=1}^h \subset \mathbb{R}^d$: $\boldsymbol{W}_i = \sum_{j \in \text{supp}(\mathbf{c}_i)} \boldsymbol{C}_{ij}\, \mathbf{w}_j\mathbf{w}_j^\top$, where $\mathbf{c}_i$ is sparse and fixed per composite $\boldsymbol{W}_i$.

The three hypotheses are nested. $\mathbf{c}_i = \mathbf{e}_i$ recovers an atomic dictionary, and a dense $\mathbf{c}_i$ drops the rank constraint, so the distributions they admit satisfy atomic $\subset$ composite $\subset$ quadratic.

Substituting this into the quadratic hypothesis gives $\mathbf{x}\mathbf{x}^\top = \sum_{i \in \text{supp}(\mathbf{z})} \boldsymbol{z}_i \sum_{j \in \text{supp}(\mathbf{c}_i)} \boldsymbol{C}_{ij}\, \mathbf{w}_j\mathbf{w}_j^\top$. The representation is reconstructed through two levels of sparsity, a dynamic selection $\mathbf{z}$ of composites, which in turn are a static selection $\mathbf{c}_i$ of base atoms. This is similar to hierarchical dictionary methods (Luo et al., 2026), but differs in that the grouping comes from a single global sparsity penalty on $\boldsymbol{C}$ instead of an added loss term. This lets the optimiser allocate capacity unevenly across composites.

**Receptive fields.** A composite encoder computes a weighted sum of squared projections, or equivalently, the expectation value of a low-rank mixture of atomic projectors:

$$\boldsymbol{z}_i^{\text{comp}} \;=\; \mathbf{x}^\top \boldsymbol{W}_i \mathbf{x} \;=\; \sum_j \boldsymbol{\lambda}_j^{(i)} \big(\mathbf{v}_j^{(i)\top} \mathbf{x}\big)^2, \qquad \boldsymbol{W}_i \;=\; \sum_{j \in \text{supp}(\mathbf{c}_i)} \boldsymbol{C}_{ij}\, \mathbf{w}_j\mathbf{w}_j^\top, \tag{4}$$

[3] Although $\mathbf{l}_i, \mathbf{r}_i$ need not be equal in practice (section 2), this is without loss of generality. Since $\mathbf{x}^\top A \mathbf{x} = \mathbf{x}^\top \frac{A+A^\top}{2} \mathbf{x}$ for any $A$, only the symmetric part $\boldsymbol{W}_i = \frac{1}{2}(\mathbf{l}_i\mathbf{r}_i^\top + \mathbf{r}_i\mathbf{l}_i^\top)$ contributes to $\boldsymbol{z}_i$. This has rank at most 2 and its eigendecomposition reduces to a signed sum of two rank-1 atoms, i.e. the composite case of subsection 3.3.

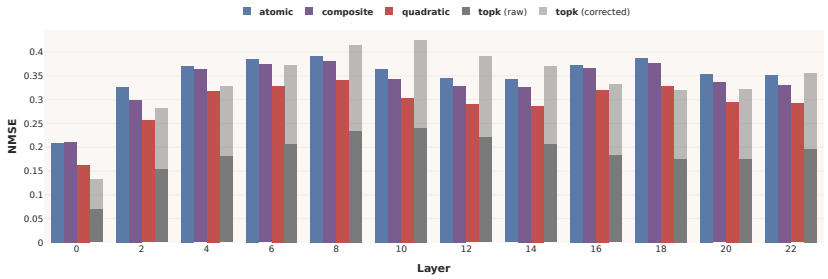


Figure 4: Reconstruction error across even layers. From layer 2 onward, the priors follow the same error ordering, `atomic` > `composite` > `quadratic`, indicating that composite latents capture more structure than their single-rank counterparts. We use a parameter-matched TopK ($K = 32$, $2\times$ latents) baseline (Gao et al., 2024), whose error is corrected for the lifted space (Appendix F).

where $\boldsymbol{\lambda}_j^{(i)}, \mathbf{v}_j^{(i)}$ are the eigenvalues and eigenvectors of $\boldsymbol{W}_i$. When $\boldsymbol{W}_i$ is semidefinite, it aggregates slab detectors into a single quadratic form whose receptive field is an *ellipsoidal region* with principal axes given by the eigenvectors of $\boldsymbol{W}_i$ and radii set by the eigenvalues. When $\boldsymbol{W}_i$ is indefinite, it combines slab detectors with opposite signs, and the receptive field becomes a *hyperboloidal region* whose positive sheet aligns with the positive eigenspace and the negative sheet with the negative eigenspace. More generally, a composite atom measures *how close to a quadric surface* $\mathbf{x}$ *is*. The data assumption is *low-rank quadratic separability*: a concept is detected through the projection of $\mathbf{x}$ onto a small subspace determined by the eigenvectors of $\boldsymbol{W}_i$ (Figure 3).

A rank-$r$ composite can achieve compact subspace capture (Bhalla et al., 2026) of any manifold living in the $r$-dimensional span of its eigenvectors, something that would require $r$ separate atoms in a linear sparse autoencoder. For example, a circle needs a rank-2 composite and a sphere a rank-3 one. Whereas Bhalla et al. (2026) recover such groupings post-hoc from co-activation statistics, the sparse mixing layer $C$ discovers them during training, baking those subspaces directly into the dictionary.

## 4 Empirical Analysis of Quadratic Latents

The three hypotheses progressively weaken the constraint on $C$, increasing expressivity and decreasing interpretability. *Atomic* fixes $C = I$, making each latent a single rank-1 atom. *Composite* imposes a global top-$K$ on $C$ (allowing $0.1\%$ non-zero entries), letting the optimiser allocate atoms unevenly across composites. *Quadratic* lifts the constraint entirely, freeing latents to reach arbitrary rank. Instantiating the full third-order tensor is impractical at scale, so it is approximated with a rank-$h$ polyadic decomposition (Hitchcock, 1927). See Appendix B for the experimental setup.

The question is where this trade-off sits in practice. We measure reconstruction and interpretability across priors, the prevalence of multi-dimensional geometry under each, and the stability of the recovered structure across runs, then qualitatively assess the extracted geometries through our interactive viewer.

### 4.1 Interpretable Composite Latents Improve Reconstruction

The utility of composites shows first in reconstruction quality. Across even layers of Qwen 3.5 (0.8B), quadratic reconstructs best, composite second and atomic worst (Figure 4). This shows how much additional structure each relaxation actually recovers. Appendix I repeats the comparison on Gemma-3-1B and Llama-3.2-1B. Atomic latents are too rigid; composition recovers structure they miss, and the composite–quadratic gap suggests that even low-rank composites fall short. This is consistent with manifolds inhabiting more ambient dimensions than their intrinsic dimension requires (Olah & Batson, 2024).

This richer quadratic prior is only useful if its latents remain readable. Bilinear latents are explained from their activating contexts as accurately as TopK ones (Appendix H) and isolate concepts slightly better under sparse probing (Appendix G).

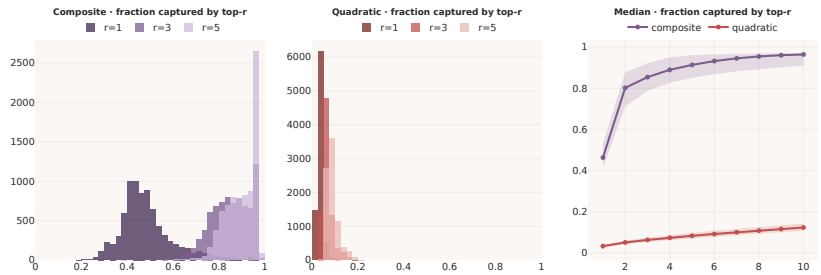


Figure 5: Comparison of the captured structure across ranks between priors. The histograms show the captured variance of all latents for the top 1, 3 and 5 dimensions while the right plot shows their median as rank increases. With sparsity applied, most latents are promoted to have very low rank with a tail of higher-dimensional structure. Without sparsity, the majority of latents are full-rank.

### 4.2 Multi-dimensional geometries are prevalent in language models

How prevalent are multi-dimensional geometries and what do they look like (Modell et al., 2025; Engels et al., 2025; Bhalla et al., 2026)? For this, we consider the spectral properties of the latents from the autoencoder on layer 18 of Qwen 3.5, taking the rank of a latent's bilinear form as a proxy for its complexity. Results depend on the structural regime: most composite latents are well-captured by 5 dimensions, while the quadratic latents are consistently high-rank, as seen in Figure 5. The former indicates that, while much structure is single-dimensional, there is a long tail of more complicated geometries that are commonly missed (Sharkey et al., 2025). The latter indicates that this structure does not emerge unless explicitly required.

Next, we turn toward a qualitative assessment of our interactive viewer from which we discuss four common geometric patterns. The observed geometry is much richer than a linear prior would predict; especially the affine nature of certain concepts is interesting (e.g., two right manifolds in Figure 1).

**Cone:** a concept spans a line or cone, the simplest geometry. 6885 activates on specific years; 7552 activates on instances of the token 'c'. The 3D visualisation often shows these latents in superposition, with non-orthogonal directions sharing a latent. **Cluster:** A subspace populated by discrete groupings without linear structure. 1972 activates on references to the USA, with small clusters around each of the tokens in 'U.S.'. 152 separates instances of "said" by context. Linear atoms could tile the point cloud but would lose the grouping. **Interpolation:** A meaningful concept that smoothly interpolates between others. 2753 puts instances of 'drug' between common addictions (coffee, smoking, alcohol). 384 activates on 'de' in names (Catherine 'de' Medici), spanning between the English 'of' (Joan 'of' Arc) and unrelated 'de' tokens. **Subset:** A latent that activates with opposite signs on distinct semantic regions of similar concepts. 190 distinguishes between occurrences of 'for'. 3021 distinguishes 'in' tokens by whether the next token is a time or location.

Together, our quantitative and qualitative results point toward an abundance of non-linear structure in the language models we studied. Dominant concepts are often approximately linear, but their surrounding geometry is not, and one-dimensional decompositions discard it. This explains why composite autoencoders achieve better reconstructions and positions them as a promising tool and lens for extracting and viewing manifolds.

### 4.3 Extracted subspaces are stable but individual latents are not

How consistent are autoencoders across runs? Existing answers rely on partial evidence: a subset of encoder directions align across runs; reconstructions correlate. Bilinear autoencoders allow an exact answer from weights alone. Each autoencoder is summarised by the kernel matrix $\boldsymbol{K} = \mathcal{W}\mathcal{W}^\top$, and two such matrices admit either a global Frobenius comparison (Equation 5) or a per-latent comparison via Hungarian matching (Equation 6), after Paulo & Belrose (2025).

$$\mathrm{Sim}_F(\boldsymbol{K}, \boldsymbol{K}') \;=\; 1 - \frac{\|\boldsymbol{K} - \boldsymbol{K}'\|_F^2}{\|\boldsymbol{K}\|_F^2 + \|\boldsymbol{K}'\|_F^2} \;=\; \frac{2\,\mathrm{Tr}(\boldsymbol{K}^\top \boldsymbol{K}')}{\|\boldsymbol{K}\|_F^2 + \|\boldsymbol{K}'\|_F^2} \tag{5}$$

$$\mathrm{Sim}_H(\mathcal{W}, \mathcal{W}') \;=\; \frac{\|\mathcal{W}^\top P \mathcal{W}'\|_F}{\|\mathcal{W}'^\top \mathcal{W}'\|_F} \quad \text{where} \quad P = \arg\max_{P \in \mathcal{P}_k} \mathrm{Tr}(\mathcal{W}^\top P \mathcal{W}') \tag{6}$$

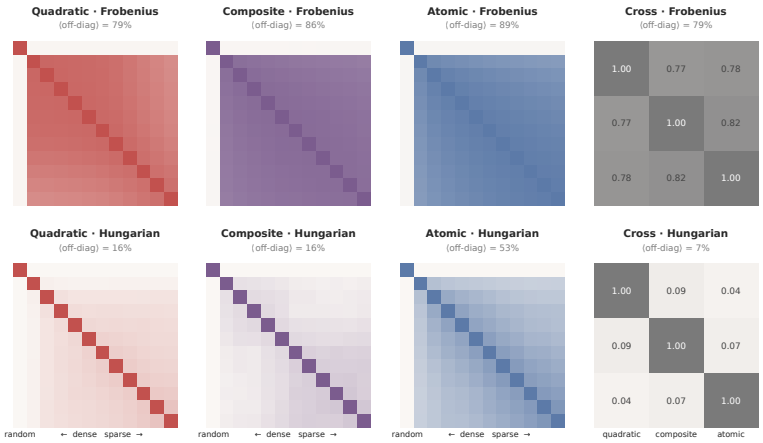


Figure 6: Frobenius (global) and Hungarian (per-latent) similarity between bilinear autoencoders with density penalties $\alpha \in [0, 1]$. Varying hyperparameters yields highly similar subspaces, yet their decompositions vary, especially for composites. The first entry is a randomly initialised autoencoder; the last column compares autoencoder kinds at $\alpha = 0.3$.

Where $\mathcal{P}_k$ denotes permutation of $k$ latents. The two metrics tell different stories. Across bilinear autoencoders that differ only in their sparsity penalty, individual latents agree only 15–50%, while the global subspace they span agrees above 90% for similar sparsity levels (Figure 6). The gap widens across architectures: individual latents agree even less, while global similarity remains high.

Current interpretability literature often discusses 'true' latents and canonical recovery (Paulo & Belrose, 2025; Leask et al., 2025; Song et al., 2025). If different autoencoders recover the same structure through different decompositions, the question becomes less about which decomposition is true and more about which is interpretable, at competitive reconstruction (Gauderis et al., 2026).

## 5 Conclusion

**Summary.** Bilinear autoencoders decompose representations into quadratic latents that admit geometric analysis from weights alone. Polynomials are an appealing non-linear primitive for interpretability: their composition stays inside the same algebraic class, and their geometry is read directly from their weight tensors. Quadratic latents are a first step beyond linear that remains tractable. section 3 formalises this through three nested hypotheses—atomic, composite, and quadratic—each a constraint on the mixing matrix $C$ that trades expressivity for interpretability. The composite regime, a global topK on $C$, recovers low-dimensional manifolds automatically, without post-hoc co-activation analysis. We find that higher-rank geometries are prevalent in language model activations, that our autoencoders recover them interpretably, and that the recovered input subspace agrees closely despite disagreement in individual atoms.

**Future work.** This paper is scoped toward geometries and their interpretable extraction and reconstruction. Hence, the most fruitful future work is to validate these latents by ablation and steering. Steering does not carry over directly: a quadratic latent offers no linear direction to add, only an energy against a quadric, so an edit moves activations toward or away from it along a landscape the weights give in closed form. Sweeping the priors, extending them to higher orders, and applying them to vision (Fel et al., 2025) are natural directions, though higher orders could grow unintelligible to humans. Constructions that order or sequence linear latents (Bussmann et al., 2025; Costa et al., 2025) could equally be built on quadratic ones. Koromilas et al. (2026) place higher-order terms only in the decoder, keeping the encoder linear. How a representation hypothesis determines the encoder and decoder structure, particularly for such asymmetric designs, remains an open question.

**Impact.** Bilinear autoencoders move beyond the linear representation hypothesis, but remain tractable both algebraically and computationally, with a well-defined description length, which is rare for nonlinear operations (Ayonrinde et al., 2024). We believe this can shift interpretability from the search for isolated features towards understandable compositional structure (Gauderis et al., 2026).

### AI USE STATEMENT

In this work, we used generative AI tools to design or provide feedback on research methodology or experiments (such as verifying correctness), methodology or experiments (primarily creating the scientific figures and interactive viewer), implementing methods (although not coding, mostly efficiency ideas) and edit a research paper to improve readability (sentence-by-sentence, to shorten or rephrase). The following do not apply to this work: generate synthetic data sets and clean and reformat dataset.

We did not use AI to help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, support qualitative and thematic data analysis, and interpret results (we did use AI for autointerp experiments, which we do not consider "interpreting results").

We manually reviewed all AI-assisted artifacts for their intended purpose.

- We visually reviewed figures and the website, although not the code.
- We read all cited papers and carefully reviewed all mathematical claims.

In brief, we used AI for small, concrete tasks while keeping the core of the method human-developed, implemented and explained. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

### ETHICS STATEMENT

This work analyses publicly available language models and corpora only. Methods that identify concept geometry aid model oversight, but could equally be used to manipulate those concepts. This is the standard dual-use risk of interpretability research. We adhere to the ICLR Code of Ethics.

### REPRODUCIBILITY STATEMENT

The architecture is defined in section 2, the full training setup in Appendix B, and pseudocode in Appendix C. Our code and interactive viewer are publicly available at [see supplementary material]and `https://bae-9xf.pages.dev`, respectively.

## REFERENCES

Michal Aharon, Michael Elad, and Alfred Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11): 4311–4322, 2006. doi: 10.1109/TSP.2006.881199.

Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L. Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. Circuit tracing: Revealing computational graphs in language models, 2025. URL `https://transformer-circuits.pub/2025/attribution-graphs/methods.html`.

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

Usha Bhalla, Thomas Fel, Can Rager, Sheridan Feucht, Tal Haklay, Daniel Wurgaft, Siddharth Boppana, Matthew Kowal, Vasudev Shyam, Jack Merullo, Atticus Geiger, and Ekdeep Singh Lubana. Do sparse autoencoders capture concept manifolds?, 2026. URL `https://arxiv.org/abs/2604.28119`.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina

Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Sam Buchanan, Druv Pai, Peng Wang, and Yi Ma. *Learning Deep Representations of Data Distributions*. Online, 2025.

Bart Bussmann, Patrick Leask, and Neel Nanda. Batchtopk sparse autoencoders, 2024. URL https://arxiv.org/abs/2412.06410.

Bart Bussmann, Noa Nabeshima, Adam Karvonen, and Neel Nanda. Learning Multi-Level Features with Matryoshka Sparse Autoencoders. In *Forty-Second International Conference on Machine Learning*, June 2025.

David Chanin, James Wilken-Smith, Tomáš Dulka, Hardik Bhatnagar, Satvik Golechha, and Joseph Bloom. A is for absorption: Studying feature splitting and absorption in sparse autoencoders, 2025. URL https://arxiv.org/abs/2409.14507.

Bob Coecke. Compositionality as we see it, everywhere around us, 2021. URL https://arxiv.org/abs/2110.05327.

Bob Coecke and Aleks Kissinger. Picturing Quantum Processes. In Peter Chapman, Gem Stapleton, Amirouche Moktefi, Sarah Perez-Kriz, and Francesco Bellucci (eds.), *Diagrammatic Represent. Inference*, Lecture Notes in Computer Science, pp. 28–31, Cham, 2018. Springer International Publishing. ISBN 978-3-319-91376-6. doi: 10.1007/978-3-319-91376-6_6.

Valérie Costa, Thomas Fel, Ekdeep Singh Lubana, Bahareh Tolooshams, and Demba Ba. From Flat to Hierarchical: Extracting Sparse Representations with Matching Pursuit, June 2025.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models, 2023. URL https://arxiv.org/abs/2309.08600.

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks, 2017. URL https://arxiv.org/abs/1612.08083.

Thomas Dooms, Ward Gauderis, Geraint Wiggins, and Jose Antonio Oramas Mogrovejo. Compositionality Unlocks Deep Interpretable Models. In *Connecting Low-Rank Representations in AI: At the 39th Annual AAAI Conference on Artificial Intelligence*, November 2024.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition, 2022. URL https://arxiv.org/abs/2209.10652.

Joshua Engels, Eric J. Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. Not all language model features are one-dimensionally linear, 2025. URL https://arxiv.org/abs/2405.14860.

Thomas Fel, Binxu Wang, Michael A. Lepori, Matthew Kowal, Andrew Lee, Randall Balestriero, Sonia Joseph, Ekdeep S. Lubana, Talia Konkle, Demba Ba, and Martin Wattenberg. Into the Rabbit Hull: From Task-Relevant Concepts in DINO to Minkowski Geometry, October 2025. URL http://arxiv.org/abs/2510.08638. arXiv:2510.08638 [cs].

Thomas Fel, Matthew Kowal, Mozes Jacobs, Dron Hazra, Usha Bhalla, Lee Sharkey, Lucius Bushnaq, Satchel Grant, Tal Haklay, Thomas Icard, Can Rager, Michael Pearce, Daniel Wurgaft, Aiden Swann, Fenil Doshi, Siddharth Boppana, Curt Tigges, Nick Cammarata, Thomas Serre, Vasudev Shyam, Owen Lewis, Thomas McGrath, Jack Merullo, Ekdeep Singh Lubana, and Atticus Geiger. Structuring sparsity: Block-sparse featurizers capture visual concept manifolds, 2026. URL https://arxiv.org/abs/2606.25234.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders, 2024. URL https://arxiv.org/abs/2406.04093.

Ward Gauderis, Thomas Dooms, Steven T. Homer, Kola Ayonrinde, and Geraint A. Wiggins. From Mechanistic to Compositional Interpretability. In *Proceedings of the 2nd Workshop on Compositional Learning: Safety, Interpretability, and Agents*. ICML, July 2026.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily

Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Peter D. Grünwald. *The Minimum Description Length Principle*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2007. ISBN 978-0-262-07281-6. Foreword by Jorma Rissanen.

Sai Sumedh R. Hindupur, Ekdeep Singh Lubana, Thomas Fel, and Demba Ba. Projecting assumptions: The duality between sparse autoencoders and concept geometry, 2025. URL https://arxiv.org/abs/2503.01822.

G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks, 2006. URL https://www.science.org/doi/abs/10.1126/science.1127647.

Frank L. Hitchcock. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6(1-4):164–189, 1927. doi: https://doi.org/10.1002/sapm192761164. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/sapm192761164.

Patrik O. Hoyer. Non-negative matrix factorization with sparseness constraints. *J. Mach. Learn. Res.*, 5:1457–1469, December 2004. ISSN 1532-4435.

Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, and Francis Bach. Proximal methods for hierarchical sparse coding. *Journal of Machine Learning Research*, 12:2297–2334, 2011.

Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Arthur Conmy, Callum McDougall, Kola Ayonrinde, Matthew Wearden, Samuel Marks, and Neel Nanda. Saebench: A comprehensive benchmark for sparse autoencoders in language model interpretability. *arXiv preprint arXiv:2503.09532*, 2025.

Panagiotis Koromilas, Andreas D. Demou, James Oldfield, Yannis Panagakis, and Mihalis Nicolaou. PolySAE: Modeling Feature Interactions in Sparse Autoencoders via Polynomial Decoding, February 2026.

Quoc Le, Alexandre Karpenko, Jiquan Ngiam, and Andrew Ng. Ica with reconstruction cost for efficient overcomplete feature learning. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper_files/paper/2011/file/233509073ed3432027d48b1a83f5fbd2-Paper.pdf.

Patrick Leask, Bart Bussmann, Michael Pearce, Joseph Bloom, Curt Tigges, Noura Al Moubayed, Lee Sharkey, and Neel Nanda. Sparse autoencoders do not find canonical units of analysis, 2025. URL https://arxiv.org/abs/2502.04878.

Michael S. Lewicki and Terrence J. Sejnowski. Learning overcomplete representations. *Neural Computation*, 12(2):337–365, 2000. doi: 10.1162/089976600300015826.

Martha Lewis. Compositional Hyponymy with Positive Operators. In *Proceedings - Natural Language Processing in a Deep Learning World*, pp. 638–647. Incoma Ltd., Shoumen, Bulgaria, October 2019. ISBN 978-954-452-056-4. doi: 10.26615/978-954-452-056-4_075. URL https://acl-bg.org/proceedings/2019/RANLP2019/pdf/RANLP075.pdf.

Yifan Luo, Yang Zhan, Jiedong Jiang, Tianyang Liu, Mingrui Wu, Zhennan Zhou, and Bin Dong. From Atoms to Trees: Building a Structured Feature Forest with Hierarchical Sparse Autoencoders, February 2026. URL http://arxiv.org/abs/2602.11881. arXiv:2602.11881 [cs].

Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, pp. 689–696, 2009. doi: 10.1145/1553374.1553463.

Alexander Modell, Patrick Rubin-Delanchy, and Nick Whiteley. The origins of representation manifolds in large language models, 2025. URL https://arxiv.org/abs/2505.18235.

Maxime Méloux, Silviu Maniu, François Portet, and Maxime Peyrard. Everything, everywhere, all at once: Is mechanistic interpretability identifiable?, 2025. URL https://arxiv.org/abs/2502.20914.

Hien Van Nguyen, Vishal M. Patel, Nasser M. Nasrabadi, and Rama Chellappa. Kernel dictionary learning. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2021–2024, Kyoto, Japan, March 2012. IEEE. ISBN 978-1-4673-0046-9 978-1-4673-0045-2 978-1-4673-0044-5. doi: 10.1109/ICASSP.2012.6288305. URL http://ieeexplore.ieee.org/document/6288305/.

Chris Olah. Mechanistic interpretability, variables, and the importance of interpretable bases, jun 27 2022. URL https://transformer-circuits.pub/2022/mech-interp-essay/index.html. Transformer Circuits Thread informal note.

Chris Olah and Joshua Batson. What is a linear representation? what is a multidimensional feature? Transformer Circuits Thread, Circuits Updates – July 2024, July 2024. URL https://transformer-circuits.pub/2024/july-update/index.html.

Bruno A. Olshausen and David J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996. doi: 10.1038/381607a0.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models, 2024. URL https://arxiv.org/abs/2311.03658.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019. URL https://arxiv.org/abs/1912.01703.

Gonçalo Paulo, Alex Mallen, Caden Juang, and Nora Belrose. Automatically interpreting millions of features in large language models. *arXiv preprint arXiv:2410.13928*, 2024.

Gonçalo Paulo and Nora Belrose. Sparse autoencoders trained on the same data learn different features, 2025. URL https://arxiv.org/abs/2501.16615.

Michael T. Pearce, Thomas Dooms, Alice Rigg, Jose M. Oramas, and Lee Sharkey. Bilinear mlps enable weight-based mechanistic interpretability, 2025. URL https://arxiv.org/abs/2410.08417.

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

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

Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978. doi: 10.1016/0005-1098(78)90005-5.

Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, Stella Biderman, Adria Garriga-Alonso, Arthur Conmy, Neel Nanda, Jessica Rumbelow, Martin Wattenberg, Nandi Schoots, Joseph Miller, Eric J. Michaud, Stephen Casper, Max Tegmark, William Saunders, David Bau, Eric Todd, Atticus Geiger, Mor Geva, Jesse Hoogland, Daniel Murfet, and Tom McGrath. Open Problems in Mechanistic Interpretability, January 2025.

Noam Shazeer. Glu variants improve transformer, 2020. URL https://arxiv.org/abs/2002.05202.

Xiangchen Song, Aashiq Muhamed, Yujia Zheng, Lingjing Kong, Zeyu Tang, Mona T. Diab, Virginia Smith, and Kun Zhang. Position: Mechanistic interpretability should prioritize feature consistency in saes, 2025. URL https://arxiv.org/abs/2505.20254.

Denis Sutter, Julian Minder, Thomas Hofmann, and Tiago Pimentel. The non-linear representation dilemma: Is causal abstraction enough for mechanistic interpretability?, 2025. URL https://arxiv.org/abs/2507.08802.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini,

Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Qwen Team. Qwen3.5: Accelerating productivity with native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Sean Tull, Robin Lorenz, Stephen Clark, Ilyas Khan, and Bob Coecke. Towards Compositional Interpretability for XAI, June 2024.

## A Notation

Scalars (0-order tensors) are written as $s$ (lower-case), vectors (1-order tensors) as $\mathbf{v}$ (bold lower-case), matrices (2-order tensors) as $M$ (upper-case) and higher-order tensors as $\mathbf{T}$ (bold upper-case). Indexed tensors are denoted as $\mathbf{T}_{ijk}$ and slicing along tensor dimensions as $\mathbf{T}_{m:n}$. We use colour to distinguish two roles that symbols play throughout this paper. Red denotes **weight-based** quantities: parameters fixed after training, analysable without any input. Blue denotes **activation-based** quantities: values that depend on an input $\mathbf{x}$ and change with each forward pass.

**Dimensions.**

- $d$ — model (input) dimension.
- $h$ — hidden dimension inside the bilinear encoder, indexed by $j$.
- $k$ — number of latents (dictionary size), indexed by $i$.
- $n$ — effective batch size (samples $\times$ sequence length).

**Activations.**

- $\mathbf{x} \in \mathbb{R}^d$ — input activation vector.
- $\boldsymbol{X} = \mathbf{x}\mathbf{x}^\top \in \mathbb{R}^{d\times d}$ — lifted product-space representation; rank-1 positive-semidefinite.
- $\hat{\boldsymbol{X}} \in \mathbb{R}^{d\times d}$ — reconstructed product-space representation.
- $\mathbf{z} \in \mathbb{R}^k$ — latent activation vector; the sparse code produced by the encoder.
- $\boldsymbol{z}_i = \mathbf{x}^\top \boldsymbol{W}_i \mathbf{x} = \langle \boldsymbol{W}_i, \boldsymbol{X} \rangle_F$ — scalar activation of the $i$-th latent.

**Encoder/Decoder weights.**

- $\boldsymbol{L}, \boldsymbol{R} \in \mathbb{R}^{h \times d}$ — left and right encoder factor matrices; $j$-th rows $\mathbf{l}_j, \mathbf{r}_j \in \mathbb{R}^d$.
- $\boldsymbol{C} \in \mathbb{R}^{k \times h}$ — composition matrix; $i$-th row $\mathbf{c}_i$ mixes hidden units into latent $i$; entry $\boldsymbol{C}_{ij}$ is the static weight of hidden unit $j$ in latent $i$.
- $\boldsymbol{W}_i = \sum_{j \in \mathrm{supp}(\mathbf{c}_i)} \boldsymbol{C}_{ij}\, \mathbf{l}_j \mathbf{r}_j^\top \in \mathbb{R}^{d \times d}_{\mathrm{sym}}$ — symmetric bilinear form of the $i$-th latent. Since $\mathbf{x}^\top A \mathbf{x} = \mathbf{x}^\top \frac{A + A^\top}{2} \mathbf{x}$, only the symmetric part of any bilinear form contributes, so $\boldsymbol{W}_i$ is taken symmetric without loss of generality.
- $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{k \times (d \times d)}$ — full encoder tensor; slice $\boldsymbol{\mathcal{W}}_i = \boldsymbol{W}_i$.
- $\boldsymbol{K} = \boldsymbol{\mathcal{W}}\boldsymbol{\mathcal{W}}^\top = \boldsymbol{C}^\top (\boldsymbol{L}\boldsymbol{L}^\top \odot \boldsymbol{R}\boldsymbol{R}^\top)\boldsymbol{C} \in \mathbb{R}^{k \times k}$ — kernel (Gram) matrix; allows evaluating the reconstruction loss without materialising the $d^2$-dimensional product space.

**Dictionary atoms (representation hypotheses).**

- $\boldsymbol{W}_i \in \mathbb{R}^{d \times d}_{\mathrm{sym}}$ — $i$-th dictionary atom (quadratic hypothesis); the same symbol as the encoder bilinear form.
- $\mathbf{w}_i \in \mathbb{R}^d$ — $i$-th atomic dictionary direction; under the atomic hypothesis $\boldsymbol{W}_i = \mathbf{w}_i \mathbf{w}_i^\top$.
- Under the composite hypothesis, $\boldsymbol{W}_i = \sum_{j \in \mathrm{supp}(\mathbf{c}_i)} \boldsymbol{C}_{ij}\, \mathbf{w}_j \mathbf{w}_j^\top$; no additional letter is needed beyond $\boldsymbol{C}_{ij}$.

**Spectral quantities (weight-based).**

- $\boldsymbol{W}_i = \boldsymbol{U}_i \boldsymbol{\Lambda}_i \boldsymbol{U}_i^\top$ — eigendecomposition of the $i$-th latent's bilinear form.
- $\boldsymbol{\lambda}_j^{(i)}$, $\mathbf{v}_j^{(i)}$ — $j$-th eigenvalue and eigenvector of $\boldsymbol{W}_i$; computable from weights alone, no forward pass required.

**Similarity metrics.**

- $\mathrm{Sim}_F(\boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{W}}') = 2\,\mathrm{Tr}(\boldsymbol{K}^\top \boldsymbol{K}')/(\|\boldsymbol{K}\|_F^2 + \|\boldsymbol{K}'\|_F^2)$ — Frobenius (global) similarity between two autoencoders.
- $\mathrm{Sim}_H(\boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{W}}')$ — Hungarian (per-latent) similarity after optimal assignment.

## B EXPERIMENTAL SETUP

We use the same hyperparameters across all discussed autoencoders. Hyperparameters follow standard practice for sparse autoencoders; results are robust across the variations we tested (Appendix I).

**Data.** In section 4, we exclusively used `Qwen3.5-0.8B-Base` (Team, 2026)—a small but competent model—with input samples from the FineWeb dataset (Penedo et al., 2024). This lets us sweep over reasonably sized autoencoders and fully train on a tight compute budget. All autoencoders train in about 15 minutes on an RTX 4080. We replicate these results on other open source models, discussed in Appendix I.

**Optimiser.** We use the Muon optimiser (Jordan et al., 2024), which we have found to significantly outperform Adam, sometimes up to $5\times$ faster convergence. We use a constant learning rate schedule. We use a linear warmup for $\alpha$ to prioritise reconstruction over activation sparsity near the start.

**Sparsity.** We use top-$K$ annealing to sparsify $\boldsymbol{C}$, leaving only $0.1\%$ of entries active: we decay to the target sparsity over the first half of training and freeze the sparsity mask for the last $20\%$, which stabilises convergence. We weigh the entries of $\boldsymbol{C}$ by the scale in $\boldsymbol{L}$ and $\boldsymbol{R}$ when doing top-$K$ to ensure the selection is global, which especially helps smoothen the start of training.

**Architecture.** We have found orthogonal initialisation to perform better than Xavier initialisation. Orthogonal initialisation favours sparser solutions at a minute reconstruction cost. We use $8,192$ latents and $16,384$ hidden dimensions across models, and $\alpha = 0.3$ by default. Weight-tying the encoder and decoder does not impact reconstruction and hence do so.

| | | | | | |
|---|---|---|---|---|---|
| **Batch size** | 32 | **Optimiser** | Muon | $d$ **(model)** | 1024 |
| **Sequence length** | 256 | **Momentum** | 0.95 | $k$ **(latents)** | $8 \times 1024$ |
| **Steps** | $2^{11}$ | **Learning rate** | 0.03 | $h$ **(hidden)** | $16 \times 1024$ |
| **Total tokens** | $\approx 16$M | **Schedule** | constant | **Default** $K$ | $0.1\%$ |
| **Shuffling** | No | $\alpha$ **warmup** | 256 steps | **Default** $\alpha$ | 0.3 |

Figure 7: Hyperparameters related to data, the optimiser, and the architecture, respectively.

Finally, our code is publicly available at [see supplementary material]and extensively uses PyTorch (Paszke et al., 2019).

## C Pseudocode

We provide distilled pseudo code for the initialisation and forward pass of the 'atomic' autoencoder. The intuition for why this works is provided in Appendix D.

```
def __init__(self, config):
    self.left = nn.init.orthogonal(config.d_features, config.d_model)
    self.right = nn.init.orthogonal(config.d_features, config.d_model)

def forward(self, acts: Tensor['batch dims'], alpha: float):
    # Compute the latents
    latents = linear(acts, self.left) * linear(acts, self.right)

    # Compute the density and regularisation term
    density = hoyer(latents, dim=0).mean()

    # Compute constituent parts of the reconstruction error
    kernel = (self.left @ self.left.T) * (self.right @ self.right.T)
    recons = einsum(latents, latents, kernel, "... h1, ... h2, h1 h2 ->
    ...")
    cross = latents.square().sum(-1)

    # Compute the error and loss
    target = acts.square().sum(-1, keepdim=True)
    error = ((recons - 2 * cross + target) / target).mean()
    loss = error + alpha * density
    return loss, dict(error=error, density=density)
```

## D Kernel trick for efficient training

Bilinear autoencoders differ from ordinary ones because they reconstruct the product space. While the inputs are naturally factored as $\boldsymbol{X} = \mathbf{x}\mathbf{x}^\top$, the outputs are not: $\hat{\boldsymbol{X}} \neq \hat{\mathbf{x}}\hat{\mathbf{x}}^\top$. Hence, naively computing the reconstruction loss requires materialising the quadratic space, which is unmanageable at scale. However, this can be avoided by expanding the numerator (using $\mathbf{z} := \mathcal{W}\boldsymbol{X}$):

$$\text{NMSE}(\hat{\boldsymbol{X}}, \boldsymbol{X}) = \frac{\|\hat{\boldsymbol{X}} - \boldsymbol{X}\|_F^2}{\|\boldsymbol{X}\|_F^2} = \frac{\mathbf{z}^\top \boldsymbol{K} \mathbf{z} - 2\|\mathbf{z}\|^2 + \|\boldsymbol{X}\|_F^2}{\|\boldsymbol{X}\|_F^2} \tag{7}$$

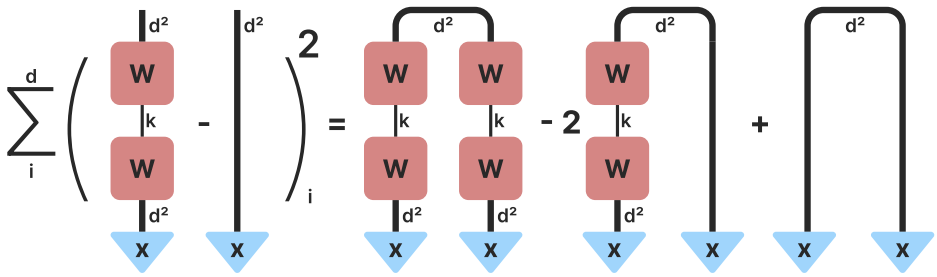


Figure 8: Diagrammatic formulation of Equation 7. Lines indicate tensor contractions over that index, while unconnected lines represent unmodified open indices. Thick lines represent a quadratic space.

The reconstruction $\hat{\mathbf{X}}$ never appears. All three terms are inner products of $\mathbf{x}$ and $\mathbf{z}$, and the first is weighted by $\boldsymbol{K}$, which is efficiently computable via Equation 8. This is reminiscent of the kernel trick to simulate an intractable polynomial feature space through inner products, except that here $\boldsymbol{K}$ is learned by the autoencoder. Due to the factorisation of $\mathcal{W}$, we can compute $\boldsymbol{K}$ without materialising any higher-order tensors:

$$\boldsymbol{K} \;=\; \mathcal{W}\mathcal{W}^{\top} \;=\; \boldsymbol{C}^{\top}(\boldsymbol{L}\boldsymbol{L}^{\top} \odot \boldsymbol{R}\boldsymbol{R}^{\top})\boldsymbol{C} \tag{8}$$

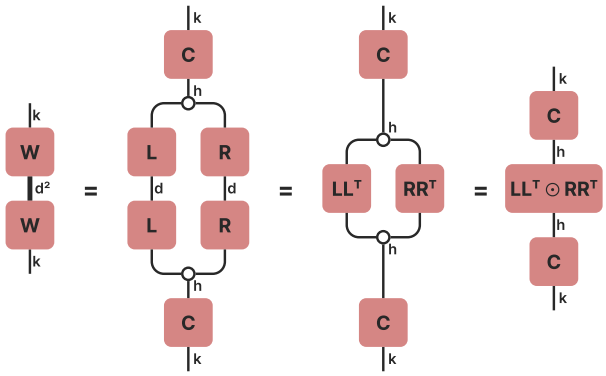


Figure 9: Diagrammatic equation for computing the kernel $\boldsymbol{K} = \mathcal{W}\mathcal{W}^{\top}$ efficiently using a sensible contraction order. The kernel matrix of a bilinear autoencoder can be evaluated by recasting it as an element-wise multiplication of two matrices.

While this trick reduces computational complexity, the kernel matrix $\boldsymbol{K} \in \mathbb{R}^{k\times k}$ can still be a memory bottleneck. However, it can be evaluated by splitting the $k \times k$ space into $p$ blocks $s_1 \oplus s_2 \oplus \cdots \oplus s_p$, which avoids full materialisation:

$$\mathbf{z}^{\top}\boldsymbol{K}\mathbf{z} = \sum_{s_a, s_b} \mathbf{z}_{s_a}\, \boldsymbol{K}_{s_a s_b}\, \mathbf{z}_{s_b}$$

This can be further improved since only the symmetric part of $\boldsymbol{K}$ contributes; sampling only the lower or upper triangular part speeds up computation by a factor of 2.

## E Proof of Theorem 1

*Proof of Theorem 1.* Since $\mathbf{x} \sim \mathrm{Unif}(S^{d-1})$ and $\mathbf{w} \in S^{d-1}$, the Poincaré–Maxwell–Borel lemma gives $\sqrt{d}\,(\mathbf{w}^{\top}\mathbf{x}) \xrightarrow{\mathcal{D}} Z \sim \mathcal{N}(0,1)$ as $d \to \infty$. Together with concentration of $\|\mathbf{g}\|$ around $\sqrt{d}$ for $\mathbf{g} \sim \mathcal{N}(0, I_d)$, this yields, for fixed $\tau$,

$$\mathbb{P}(\mathbf{w}^{\top}\mathbf{x} > \tau) \;\sim\; \mathbb{P}(Z > \tau\sqrt{d}), \qquad \mathbb{P}\big((\mathbf{w}^{\top}\mathbf{x})^2 > \tau\big) \;\sim\; 2\,\mathbb{P}(Z > \sqrt{\tau d}).$$

Applying the standard Gaussian tail asymptotic via Mills' ratio, $\mathbb{P}(Z > a) = \Theta(a^{-1}e^{-a^2/2})$ for $a > 0$:

$$\frac{\mathbb{P}\big((\mathbf{w}^\top \mathbf{x})^2 > \tau\big)}{\mathbb{P}(\mathbf{w}^\top \mathbf{x} > \tau)} = \frac{\Theta\big((\tau d)^{-1/2}\, e^{-\tau d/2}\big)}{\Theta\big((\tau^2 d)^{-1/2}\, e^{-\tau^2 d/2}\big)} = \Theta\Big(\sqrt{\tau}\, e^{-d\tau(1-\tau)/2}\Big) = \Theta\Big(e^{-d\tau(1-\tau)/2}\Big),$$

where the last equality absorbs the fixed constant $\sqrt{\tau}$ into the $\Theta$. Since $\tau \in (0, 1)$, the exponent is strictly negative, confirming exponential decay in $d$. □

## F Product space error correction

This section discusses the discrepancy between reconstructing $\mathbf{x}$ and $\mathbf{x}\mathbf{x}^\top$. This product-space correction admits an exact closed form. Taking $\|\mathbf{x}\|^2 = 1$ due to scale-invariance (Equation 7) and substituting $\mathbf{x}^\top \hat{\mathbf{x}} = \frac{1}{2}(1 + \|\hat{\mathbf{x}}\|^2 - s)$ with $s = \|\mathbf{x} - \hat{\mathbf{x}}\|^2$,

$$\begin{aligned} S &= \|\mathbf{x}\mathbf{x}^\top - \hat{\mathbf{x}}\hat{\mathbf{x}}^\top\|_F^2 \\ &= 1 + \|\hat{\mathbf{x}}\|^4 - 2\,(\mathbf{x}^\top \hat{\mathbf{x}})^2 \\ &= \tfrac{1}{2}(1 - \|\hat{\mathbf{x}}\|^2)^2 + s(1 + \|\hat{\mathbf{x}}\|^2) - \tfrac{1}{2}s^2. \end{aligned} \tag{9}$$

Under the mild assumption $\|\hat{\mathbf{x}}\|^2 \approx 1$, which any sensible reconstruction should satisfy, the first term vanishes and the third is negligible for small $s$, leaving $S \approx 2s$. The product-space loss is therefore approximately twice the input-space loss. We use the exact correction in Figure 4, which places the baseline in the same space as our models. Reconstruction is then comparable across variants and layers, suggesting that bilinear autoencoders can leverage decodable structure in interactions between input elements.

## G Probing on autoencoder latents

We follow the setup of SAEBench (Karvonen et al., 2025; Koromilas et al., 2026): 36 tasks over six labelled datasets (professions, product categories, review sentiment, programming languages, news topics, languages), 4000 training and 1000 test texts per task, balanced, each text its first 128 tokens. Layer-18 activations of Qwen3.5-0.8B and all latent values are averaged over the text. Per task, the latent that best separates the two classes on the training texts is selected and a logistic probe on it is scored on the test texts. A TopK latent is probed on its pre-activation. A bilinear latent is probed on its subspace (pre-multiplication); the equivalent of its pre-activation.

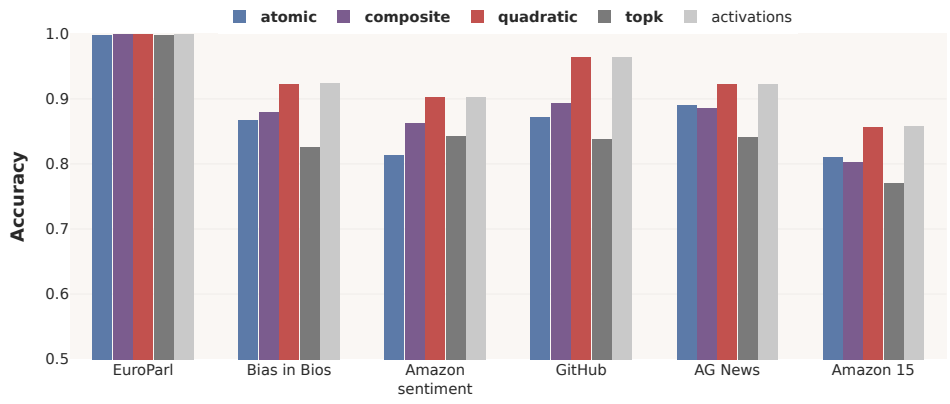


Figure 10: Probing accuracy of the best single latent per dictionary on each SAEBench dataset; grey is a linear probe on the averaged activations. We probe the pre-activations of all latents, restricted to 2, 5, and 20 dimensions for atomic, composite, and quadratic latents, respectively. This paints a similar picture to Figure 4: adding a few dimensions increases performance, indicating that most features lie on low-dimensional manifolds.

## H AUTOMATED INTERPRETABILITY

We compare explanations between autoencoder latents using autointerp scores (Paulo et al., 2024). For 200 random live latents, an explainer model (Claude Sonnet 5) writes a one-sentence explanation from 32 activating 32-token windows of held-out FineWeb-Edu text, and a scorer (also Sonnet) labels 20 additional windows, 10 in which the latent is active and 10 random, as active or not; we report the accuracy of these labels. A bilinear latent is active on a token where $|z_i - \mu_i|$ exceeds a per-latent threshold, with $\mu_i$ the mean activation of latent $i$, set so that every latent fires as often as the average latent of the TopK baseline. A TopK latent is simply active where its activation is nonzero.

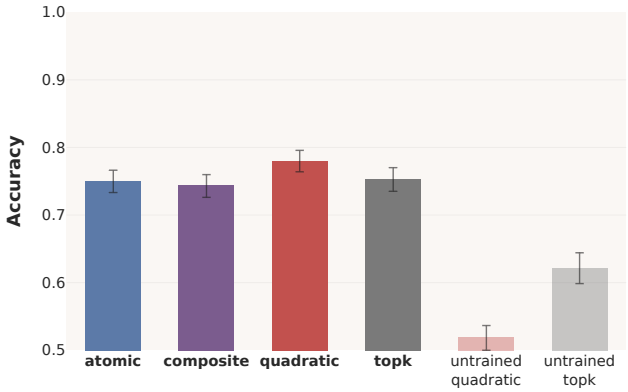


Figure 11: Detection accuracy per dictionary, mean over latents with 95% bootstrap intervals. As a floor, we use untrained dictionaries. Explanations of bilinear latents are recognised about as reliably as those of TopK latents, with the quadratic dictionary slightly ahead, and every trained dictionary sits well above its untrained counterpart.

In short, bilinear latents are explained as accurately as TopK latents, and quadratic latents score highest. A full-rank form is as explainable from its activating contexts as a low-rank one, though its geometry is harder to display.

## I RESULTS ON ADDITIONAL MODELS

We extend our empirical results from subsection 4.1 on two other well-known open-source models: Gemma-3-1B (Team et al., 2025) and Llama-3.2-1B (Grattafiori et al., 2024). These are in line with expectations from Qwen3.5-0.8B.

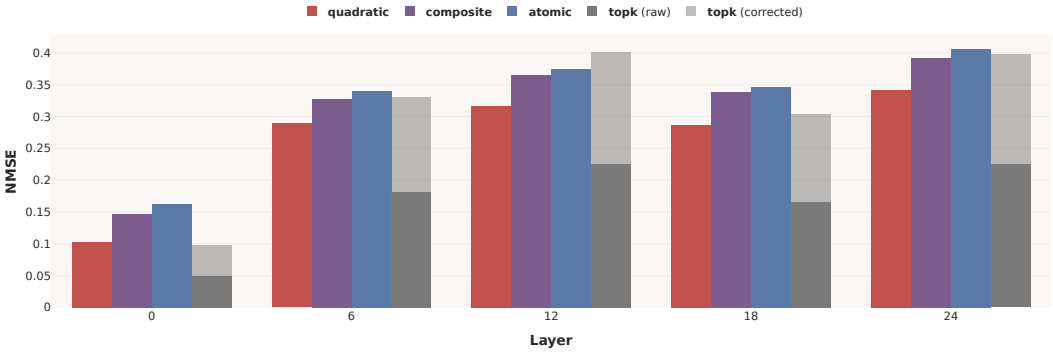


Figure 12: Sweep over selected layers on Gemma-3-1B using the hyperparameters discussed in Appendix B. To clarify, $h = 16,384$ and $k = 8,192$ (i.e. not scaled with the slightly increased model dimension of $d = 1152$). The results are in line with subsection 4.1, likely because the model width and depth resemble Qwen3.5-0.8B.

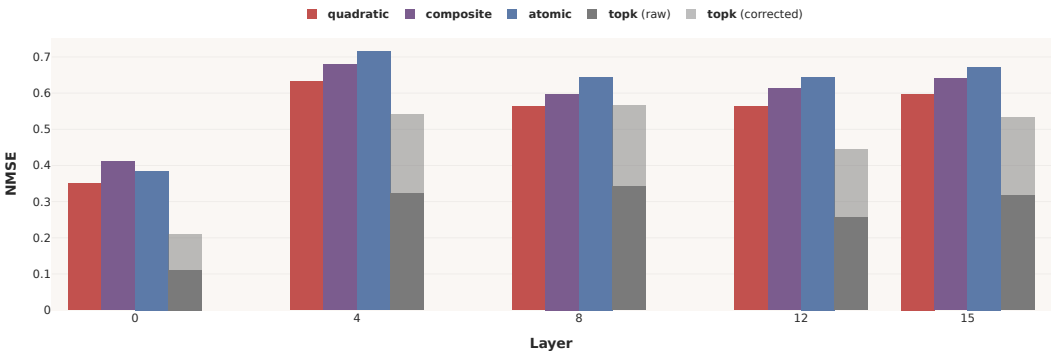


Figure 13: Sweep over selected layers on Llama-3.2-1B using the hyperparameters discussed in Appendix B. Reconstruction is much worse for both the topK baseline and our models, which is expected since $k$ and $h$ are kept constant despite $d = 2048$ being twice as large; this means the quadratic space is $4\times$ as large. As on Qwen, layer 0 is where composite trails atomic.

## J Interactive viewer

To make the extracted geometries easy to inspect, we publish an interactive viewer at `https://bae-9xf.pages.dev` covering all 8,192 latents from a composite autoencoder trained on layer 18 of Qwen3.5-0.8B with ≈64M tokens, $8\times$ longer than for Figure 4.

The landing page lays out the corpus as a scatter over four of the summary statistics defined below, with an alternative UMAP view. The 20 highest-importance latents are highlighted in a sidebar, each carrying an unevaluated 2–4 word label produced by a single Claude pass over its top-activating contexts.

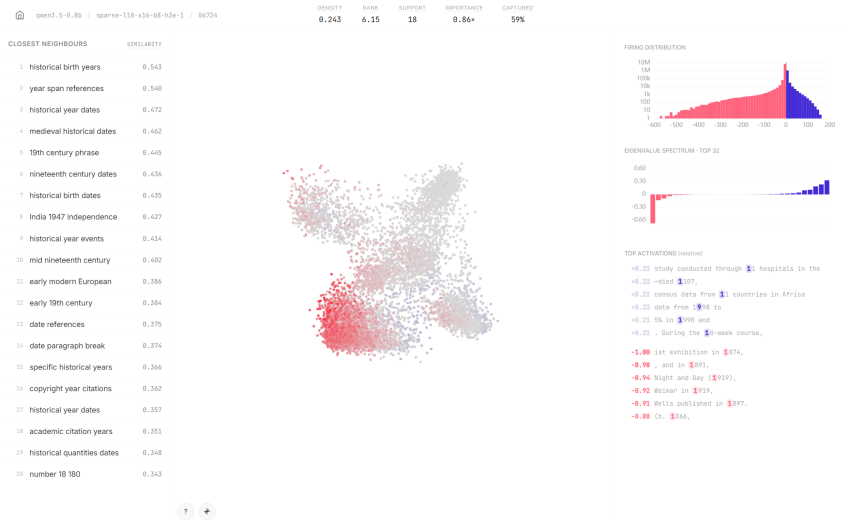


Figure 14: Screenshot of the interactive viewer user interface for a particular latent: `https://bae-9xf.pages.dev/composite/06724`.

Each composite has a dedicated page anchored by a 3D scatter of token contexts, projected onto the top-3 eigenvectors of $\boldsymbol{W}_i = \boldsymbol{U}_i \boldsymbol{\Lambda}_i \boldsymbol{U}_i^\top$ (selected by $|\boldsymbol{\lambda}|$). Eigendirections optionally overlay the scatter as bidirectional axes (purple for positive $\boldsymbol{\lambda}$, pink for negative; bidirectional because the bilinear form depends only on $(\mathbf{v}_j^{(i)\top}\mathbf{x})^2$). Points are drawn from ∼8M tokens by weighted reservoir sampling $w \propto (\|\mathbf{z}\| + \varepsilon)^4$, over-representing samples far from the origin so the manifold isn't drowned by the dense central cluster. Surrounding panels show the eigenvalue spectrum (with the three rendered axes labelled $X$, $Y$, $Z$), the 6 highest-activating $\pm 4$-token contexts per firing sign, and the 20 closest neighbours ranked by subspace overlap $\|\boldsymbol{U}_i^\top \boldsymbol{U}_j\|_F^2 / \sqrt{r_i r_j}$.

The header reports five summary statistics. Density measures how concentrated the firing distribution is. Effective rank gauges how many eigenvalues carry meaningful weight. Support counts the encoder atoms feeding into $W_i$. Importance reports the total spectral energy, normalised so the average composite reads $1\times$. Captured is the spectral share of the three rendered eigenvectors and signals how faithfully the 3D scatter represents the full manifold.

$$\text{density} = \text{Hoyer}(\mathbf{z}_i), \qquad \text{importance} = \textstyle\sum_j (\lambda_j^{(i)})^2,$$

$$\text{effective rank} = \frac{(\sum_j |\lambda_j^{(i)}|)^2}{\sum_j (\lambda_j^{(i)})^2}, \qquad \text{captured} = \frac{|\lambda_1^{(i)}| + |\lambda_2^{(i)}| + |\lambda_3^{(i)}|}{\sum_j |\lambda_j^{(i)}|}.$$

## K Random viewer examples

To avoid selection bias, we show 6 random latents and their geometry.

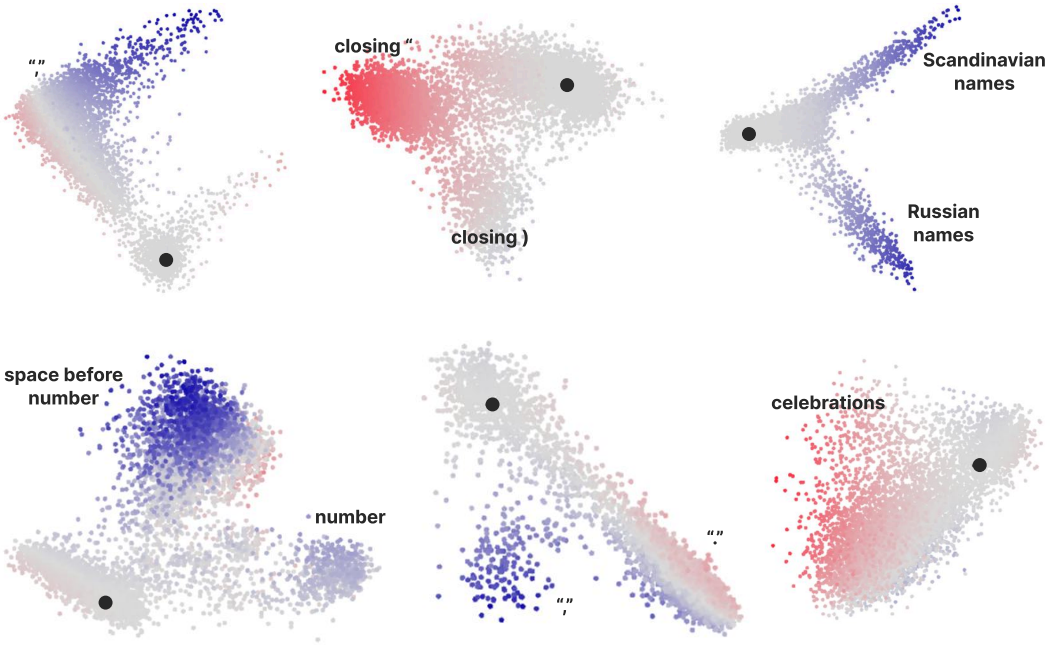


Figure 15: The first few manifolds by index from our interactive viewer (indices 1 and 4 were excluded as they are dense and uninteresting, about 10% of latents are dense). This is a representative sample for how many features are primarily linear (top-left and top-right) versus more cluster-based (the others).