From Flat to Hierarchical: Extracting Sparse Representations with Matching Pursuit

Valérie Costa[1*] **Thomas Fel**[2*] **Ekdeep Singh Lubana**[3,4*]
Bahareh Tolooshams[5,6†] **Demba Ba**[2,7†]

[1]EPFL [2]Kempner Institute, Harvard University
[3]CBS-NTT Program in Physics of Intelligence, Harvard University
[4]Physics of Artificial Intelligence Group, NTT Research, Inc., Sunnyvale, CA, USA
[5]University of Alberta [6]Alberta Machine Intelligence Institute (Amii) [7]SEAS, Harvard University

Abstract

Motivated by the hypothesis that neural network representations encode abstract, interpretable features as linearly accessible, approximately orthogonal directions, sparse autoencoders (SAEs) have become a popular tool in interpretability literature. However, recent work has demonstrated phenomenology of model representations that lies outside the scope of this hypothesis, showing signatures of hierarchical, nonlinear, and multi-dimensional features. This raises the question: do SAEs represent features that possess structure at odds with their motivating hypothesis? If not, does avoiding this mismatch help identify said features and gain further insights into neural network representations? To answer these questions, we take a construction-based approach and re-contextualize the popular matching pursuit (MP) algorithm from sparse coding to design MP-SAE—an SAE that unrolls its encoder into a sequence of residual-guided steps, allowing it to capture hierarchical and nonlinearly accessible features. Comparing this architecture with existing SAEs on a mixture of synthetic and natural data settings, we show: (i) hierarchical concepts induce conditionally orthogonal features, which existing SAEs are unable to faithfully capture, and (ii) the nonlinear encoding step of MP-SAE recovers highly meaningful features, helping us unravel shared structure in the seemingly dichotomous representation spaces of different modalities in a vision-language model, hence demonstrating the assumption that useful features are solely linearly accessible is insufficient. We also show that the sequential encoder principle of MP-SAE affords an additional benefit of adaptive sparsity at inference time, which may be of independent interest. Overall, we argue our results provide credence to the idea that interpretability should begin with the phenomenology of representations, with methods emerging from assumptions that fit it.

1 Introduction

Modern neural networks trained on large-scale datasets have achieved unprecedented performance on several practical tasks [1–10], prompting efforts towards understanding how these abilities are implemented within a model [11–13]. To this end, Sparse Autoencoders (SAEs) [14–23], motivated by the Linear Representation Hypothesis (LRH) [24, 25], have become a popular tool for interpreting neural networks. Specifically, LRH, a phenomenological model of organization and computation of neural network representations [24–28], posits that neural network representations of dimension m can be decomposed along a basis of $p \gg m$ approximately orthogonal directions that reflect abstract, interpretable concepts underlying the data distribution. Here, approximate orthogonality is argued to be necessary for circumventing the problem of packing more unique vectors (concepts) than the

[*] Equal contribution. [†] Equal advising. Emails: `btolooshams@ualberta.ca`, `demba@seas.harvard.edu`.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

dimensionality of the space allows (a.k.a. the superposition problem [24]), hence enabling reliable estimation of a concept's presence by a linear operator (e.g., the linear map in an MLP) [29, 30]. Grounded in this idea, and inspired by its relation to the well-known problem of sparse coding [31, 32], SAEs have been proposed to learn, in an unsupervised manner, a sparse, overcomplete dictionary of directions that (ideally) map onto abstract, interpretable concepts encoded in a neural network, hence enabling its interpretability [33, 14].

Despite the fact that substantial empirical support has been shown in favor of LRH—e.g., representations capturing concepts like geometry and lighting [34–38], facial characteristics [39, 40], broader scene organizations [41–43], and instance structures [44–51] have been shown in vision models, while ones capturing concepts like basic semantics [52–57], character roles and theory-of-mind properties [58, 59], arithmetic concepts [60], refusal to unsafe queries [61, 62], and subject–object relations [63–66] have been shown in language models—our goal in this work is to better contextualize the assumptions underlying the design of SAEs in light of recent evidence *against* the validity of LRH [67, 68]. For example, Park et al. [69] recently analyzed the organization of hierarchical and categorical concepts in neural network representations, showing that cross-entropy loss tends to encourage orthogonal structure for hierarchical relations, while categorical concepts are arranged as polytopes with overlapping support. Meanwhile, Csordas et al. [70] demonstrated the existence of "onion-like" representations in a simple copying task that cannot be linearly accessed; the existence of such "nonlinear" representations in larger-scale models was also emphasized by Engels et al. [71]. Finally, the existence of multi-dimensional features for concepts with temporal relations, e.g., days-of-the-week, has been used to argue against the use of directions as a basis for decomposing neural network representations [72, 21]; this is also supported by experiments showing concepts' representations and dimensionality can be flexibly manipulated via the inputted context [73].

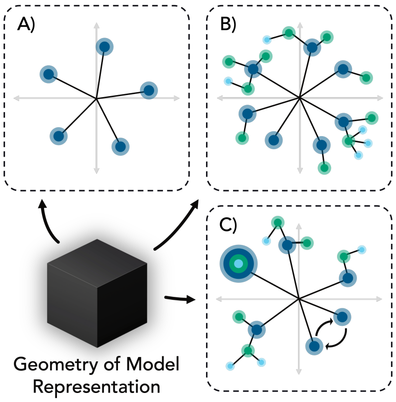


Figure 1: **Conceptual organization in neural representations. A)** *Linearly accessible concepts*: abstract directions that are approximately orthogonal and independently interpretable, as assumed by the Linear Representation Hypothesis (LRH). **B)** *Hierarchical concepts*: representations structured in parent–child relations. **C)** *Nonlinear, multidimensional, and temporally structured concepts*: features that cannot be accessed via a single direction.

The disparity highlighted above between phenomenology of neural network representations identified in recent work versus one hypothesized by LRH raises the question as to whether SAEs, which were motivated by LRH, are able to capture and explain representations that lie outside the scope of LRH (Fig. 1). Since the SAE architecture is not explicitly biased against such representations, is it possible SAEs remain performant towards expressing them, or do these disparities manifest as limitations on what concepts SAEs can explain? To address these questions, we make the following contributions.

- **Contextualizing SAEs beyond the linear representation hypothesis.** Via a mixture of synthetic and natural data, we analyze whether popular SAE architectures are able to capture concepts encoded in a hierarchical, nonlinearly accessible manner (see Hindupur et al. [21] for an analysis of multi-dimensional concepts). To this end, we introduce the notion of *Conditional Orthogonality* (Def. 2.3), under which, in a hierarchy, orthogonality is expected only between concepts within across different levels. Our experiments yield consistent evidence that existing SAEs, including those trained with hierarchical objectives [20], are unable to capture such concepts faithfully.
- **Introducing MP-SAE for phenomena beyond linear assumptions.** To contextualize the results above and understand if capturing hierarchical structures yields meaningful features in larger-scale models, we design an SAE architecture that can capture (certain kinds of) hierarchical and nonlinear structures. Specifically, we extend the standard SAE framework by unrolling the Matching Pursuit algorithm [74] into a residual-guided, sequential encoder—called the *MP-SAE*. Each step of MP-SAE selects a feature (approximately) orthogonal to what has already been explained, naturally promoting a conditionally orthogonal structure. Across experiments on both synthetic benchmarks and real-world vision-language models, MP-SAE proves more expressive and uncovers richer structural features than existing SAEs. These results question the assumption that meaningful features must be linearly and independently accessible.

- **Additional advantages of MP-SAE.** We find MP-SAE recovers richer structure than standard SAEs across diverse settings: on synthetic trees with controlled interference, it uniquely preserves both within- and across-level organization; on large-scale models (e.g., CLIP [75], DINOv2 [76]), it identifies nonlinear and multimodal features that elude linear encoders. Moreover, we show MP-SAE offers two practical benefits: (i) it enables adaptive sparsity without retraining—features can be incrementally added at test time—and (ii) it ensures monotonic improvement in reconstruction error with each inference step. These properties especially contrast with fixed-k SAEs, which often degrade when sparsity levels shift at test time. Thus, though our goal for analyzing MP-SAE was to understand whether useful features can be elicited by actively modeling phenomenology outside the scope of LRH, we believe MP-SAE can be of independent interest to the community.

2 Formalizing Linear Representation Hypothesis and Sparse Autoencoders

We begin by formally describing two objects central to our study: linear representation hypothesis and sparse autoencoders. We note that though the idea that concepts are "linearly encoded" in neural networks has found significant attention in literature (see App. A), a formal definition has rarely been offered (see [25, 77] for notable exceptions). Consequently, several related observations on neural network representations have come to be associated with LRH, including linear accessibility of interpretable concepts [78, 59], linear algebraic manipulability of model behavior [79, 66, 80], and decomposability into a linear mixture of directions [14, 23, 22]. Hence, to be precise about what we mean by the term, we distill the core claims posited by Elhage et al. [24] on LRH and formally state (our interpretation of) the hypothesis. This definition then directly motivates SAEs.

In what follows, we use plain lowercase letters a to denote scalars, bold lowercase letters $\boldsymbol{a}$ to denote vectors, and uppercase letters $\boldsymbol{A}$ to denote matrices. The k^{th} column of a matrix $\boldsymbol{A}$ is written as $\boldsymbol{A}_k$. The norms $\|\cdot\|_2$ and $\|\cdot\|_1$ denote the standard ℓ_2 and ℓ_1 norms, while $\|\cdot\|_0$ refers to the ℓ_0 *pseudo*-norm, i.e., the number of nonzero entries. The Frobenius norm of a matrix is denoted by $\|\cdot\|_F$. For any vector $\boldsymbol{z}$, its support is defined as $\operatorname{supp}(\boldsymbol{z}) := \{i : z_i \neq 0\}$, and $\boldsymbol{z}$ is said to be k-sparse if $|\operatorname{supp}(\boldsymbol{z})| = k$. Hereafter, we use "features" and "concepts" as interchangeable terms, distinguishing them from "representations" a model derives for an input.

Definition 2.1 (**Linear Representation Hypothesis (LRH)**). *A representation $\boldsymbol{x} \in \mathbb{R}^m$ is said to satisfy the* linear representation hypothesis *(LRH) if there exists a dictionary $\boldsymbol{D} = [\boldsymbol{D}_1, \ldots, \boldsymbol{D}_p] \in \mathbb{R}^{m \times p}$ and a coefficient vector $\boldsymbol{z} \in \mathbb{R}^p$ such that $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{z}$, under the following conditions:*

$$\begin{cases} \textbf{\textit{(i)}}\ \textit{Overcompleteness:} & p \gg m; \\ \textbf{\textit{(ii)}}\ \textit{Quasi-orthogonality:} & \max\limits_{i \neq j} \left| \boldsymbol{D}_i^\top \boldsymbol{D}_j \right| \leq \varepsilon, \quad \textit{where } \forall i \ \ \|\boldsymbol{D}_i\|_2 = 1 \ ; \ \textit{and} \\ \textbf{\textit{(iii)}}\ \textit{Sparsity:} & |\operatorname{supp}(\boldsymbol{z})| \leq k \ll p. \end{cases}$$

We emphasize the constraints above are deeply interdependent. In particular, if a model is to represent a large number of distinct concepts ($p \gg m$) within a lower-dimensional space $\mathbb{R}^m$, while ensuring that a linear readout can reliably recover any active concept, the directions associated with these concepts must be approximately orthogonal. This requirement is substantiated by a reinterpretation of the Johnson–Lindenstrauss (JL) lemma [29, 30][2]. While the lemma is typically used to show that high-dimensional data can be compressed into lower dimensions while preserving pairwise distances, Elhage et al. [24] apply a flipped version: one can embed exponentially many quasi-orthogonal directions *within* $\mathbb{R}^m$ such that sparse linear combinations remain distinguishable. This perspective justifies the feasibility of constructing overcomplete, low-coherence dictionaries in LRH. Rather than compressing structure, the goal is to expand representational capacity: to pack many interpretable directions into a shared space while preserving linear accessibility.

Overall, if the conditions above hold, sparse linear decompositions become not only possible, but a natural mechanism for expressing concept-aligned features—this leads to the idea of SAEs. Specifically, noting that the model described in Def. 2.1 closely aligns with the generative model assumed in classical work on sparse coding [32, 31, 81], wherein one seeks to express data as sparse linear combinations over an overcomplete dictionary, SAEs were recently proposed to re-contextualize that literature's tools for identifying concepts encoded in a neural network's representations [14–23].

[2] JL-lemma [29] states that $p = e^{\mathcal{O}(m)}$ vectors can be embedded in $\mathbb{R}^m$ such that all pairwise inner products are bounded by a small $\varepsilon > 0$, thereby justifying the feasibility of near-orthogonality even when $p \gg m$.

Definition 2.2 (**Sparse Autoencoders**). *Given model representations $\boldsymbol{x} \in \mathbb{R}^m$, the goal of an SAE is to compute a sparse code $\boldsymbol{z}$ that reconstructs $\boldsymbol{x}$ as a linear combination of a learned dictionary $\boldsymbol{D}$:*

$$\begin{aligned} \boldsymbol{z} &= \boldsymbol{\Pi}\{\boldsymbol{W}^\top(\boldsymbol{x} - \boldsymbol{b}_{pre}) + \boldsymbol{b}\}, \quad \textit{and} \\ \hat{\boldsymbol{x}} &= \boldsymbol{D}\boldsymbol{z} + \boldsymbol{b}_{pre}, \end{aligned} \tag{1}$$

where $\boldsymbol{W} \in \mathbb{R}^{m \times p}$ and $\boldsymbol{b} \in \mathbb{R}^p$ denote the encoder weights and bias, $\boldsymbol{b}_{pre} \in \mathbb{R}^m$ is a pre-decoding bias, and $\boldsymbol{D} \in \mathbb{R}^{m \times p}$ is the learned dictionary of concept vectors.

In the above, the nonlinearity projection $\boldsymbol{\Pi}\{\cdot\}$ projects the pre-activation to a sparse support [21]; common choices for activation maps include ReLU [23, 14], TopK [82, 15, 17], and JumpReLU [18, 19]. Training proceeds by minimizing a reconstruction loss along with sparsity and auxiliary penalties:

$$\mathcal{L} = \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2 + \lambda\,\mathcal{R}(\boldsymbol{z}) + \alpha\,\mathcal{L}_{\text{aux}},$$

where $\mathcal{R}(\boldsymbol{z})$ promotes sparsity, via ℓ_1 or ℓ_0 mechanisms, and $\mathcal{L}_{\text{aux}}$ may be used to minimize number of inactive units [15, 19].

2.1 Stress-Testing SAEs via Conditional Orthogonality—a Structure Outside LRH's Scope

While SAEs provide a natural operationalization of LRH, their effectiveness hinges on structural assumptions that may not hold in practice. In particular, Eq. 1 presumes that concepts correspond to approximately orthogonal directions and can be recovered via a linear projection. However, as several recent works show, these assumptions can prove overly rigid [72, 68, 71, 21, 69]. For instance, neural network representations are known to encode hierarchically structured concepts [83–88]. Park et al. [69] show that for such hierarchical concepts, while concepts at the same level of hierarchy frequently form polytopes with overlapping support, parent and child concepts tend to span orthogonal subspaces. In what follows, we formalize this idea under the term 'conditional orthogonality' and use it as a structure to stress-test whether SAEs can identify concepts that encoded in a manner outside the scope of LRH. We note this definition is merely a paraphrased version of the one offered by Park et al. [69].

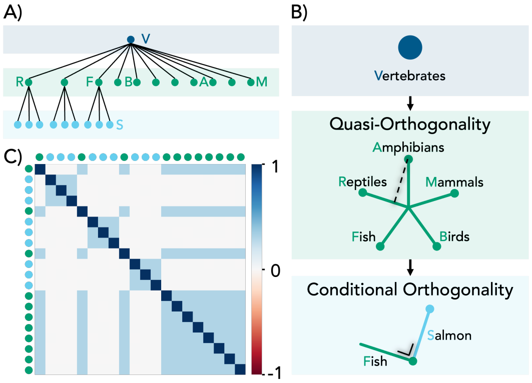


Figure 2: **Illustrative Example of Conditional vs. Quasi-Orthogonality. A)** Example of a hierarchical concept tree. **B)** Comparison of quasi-orthogonality (interference within levels) vs. conditional orthogonality (orthogonality across levels). **C)** Correlation matrix of features sampled from A, showing conditional orthogonality (white, $= 0$) across levels and quasi-orthogonality (light blue, $= \varepsilon$) within levels.

Definition 2.3 (**Conditional Orthogonality**). *Let $\boldsymbol{D} \in \mathbb{R}^{m \times p}$ be a dictionary whose columns $\boldsymbol{D}_1, \ldots, \boldsymbol{D}_p$ denote concept directions, and let $\ell : [p] \to \mathbb{N}$ assign each concept to a discrete level in a hierarchy. We say that $\boldsymbol{D}$ is* conditionally orthogonal *with respect to ℓ if*

$$\boldsymbol{D}_i^\top \boldsymbol{D}_j = 0 \quad \textit{whenever } \ell(i) \neq \ell(j).$$

Relating back to LRH (Def. 2.1), we note Def. 2.3 relaxes the global quasi-orthogonality constraint posited by LRH, requiring instead orthogonality only between concept vectors drawn from different hierarchical levels: specifically, note that no constraint is imposed on inner products $\boldsymbol{D}_i^\top \boldsymbol{D}_j$ for pairs $i, j \in [p]$ with $\ell(i) = \ell(j)$. This permits controlled interference within a given hierarchical level, while preserving separation across levels. Fig. 2, adapted from Bussmann et al. [20], depicts an instance of such structure, where hierarchical organization gives rise to distinct patterns of inter- and intra-level alignment. To evaluate whether conventional SAEs (implicitly aligned with LRH) are capable of capturing the interference structure induced by conditional orthogonality, we will analyze a synthetic generative process grounded in the taxonomy of Fig. 2 in Sec. 4.1. This will allow us to probe the inductive biases of different SAE variants under controlled structural constraints. However, to contextualize these results, we next develop an SAE that is specifically motivated to capture the ability to model features that are conditionally orthogonal in nature.

Algorithm 1 Matching Pursuit Sparse Autoencoders (MP-SAE)

```
1:  Input: Dictionary $\boldsymbol{D}$, bias $\boldsymbol{b}_{\text{pre}}$, data $\boldsymbol{x}$, steps $T$
2:  Initialize residual: $\boldsymbol{r}^{(0)} = \boldsymbol{x} - \boldsymbol{b}_{\text{pre}}$
3:  Initialize reconstruction: $\hat{\boldsymbol{x}}^{(0)} = \boldsymbol{b}_{\text{pre}}$
4:  Initialize sparse code: $\boldsymbol{z} = \mathbf{0}$
5:  for $t = 0, \ldots, T-1$ do
6:      $j^{(t)} = \arg\max_{j \in [p]} (\boldsymbol{D}^\top \boldsymbol{r}^{(t)})_j$
7:      $\boldsymbol{z}^{(t)}_{j^{(t)}} = \boldsymbol{D}^\top_{j^{(t)}} \boldsymbol{r}^{(t)}$
8:      $\hat{\boldsymbol{x}}^{(t+1)} = \hat{\boldsymbol{x}}^{(t)} + \boldsymbol{z}^{(t)}_{j^{(t)}} \boldsymbol{D}_{j^{(t)}}$
9:      $\boldsymbol{r}^{(t+1)} = \boldsymbol{r}^{(t)} - \boldsymbol{z}^{(t)}_{j^{(t)}} \boldsymbol{D}_{j^{(t)}}$
10: end for
11: Output: Sparse code $\boldsymbol{z} = \sum_{t=0}^{T-1} \boldsymbol{z}^{(t)}$, Reconstruction $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}^{(T)}$
```

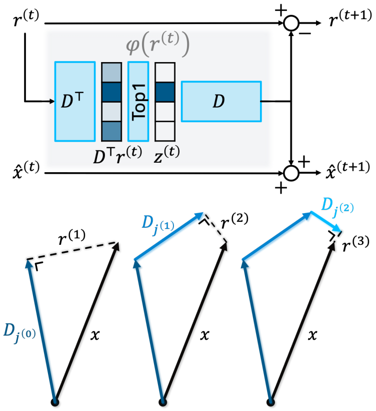


Figure 3: **(Top Left) One iteration of the MP-SAE encoder:** The residual $\boldsymbol{r}^{(t)}$ is projected onto the dictionary $\boldsymbol{D}$, the most correlated feature $\boldsymbol{D}_{j^{(t)}}$ is selected, and its contribution is subtracted from the residual and added to the reconstruction. **(Bottom Left) MP sequentially reconstructs $\boldsymbol{x}$** by greedily selecting features that best explain the residual, promoting orthogonal selection and enabling access to higher-order features that are nonlinearly accessible from $\boldsymbol{x}$. **(Right)** MP-SAE embeds the MP algorithm into a sparse autoencoder, where the dictionary $\boldsymbol{D}$ is learned through backpropagation.

3 Operationalizing Conditional Orthogonality via MP-SAE

To enable the ability to identification of features encoded in a *conditionally orthogonal* manner (Def. 2.3), we construct a sparse autoencoder whose inference reflects the structure of hierarchical representations—minimizing interference across levels while tolerating it within [69]. Our design draws on the Matching Pursuit (MP) algorithm [74] from sparse coding [89–92], which has been shown to recover features from the appropriate hierarchical level in conditionally orthogonal dictionaries [93]. MP performs inference greedily: at each step, the feature most correlated with the residual is selected, its contribution subtracted, and the process repeated. This iterative residual decomposition promotes conditionally orthogonal feature selection, as each new feature explains variance orthogonal to what was captured by the previous feature (Fig. 3). Embedding this mechanism into a sparse autoencoder yields the *Matching Pursuit Sparse Autoencoder (MP-SAE)*, formalized in Algorithm 1. Below, we unpack the structure and implications of MP-SAE, examining: (***i***) the mechanics of its inference procedure; (***ii***) how this procedure promotes conditional orthogonality; and (***iii***) its capacity to extract higher-order, nonlinearly accessible features.

(*i*) Mechanics of MP-SAE. MP-SAE uses a shared learned dictionary for encoding and decoding. The encoder embeds the classical Matching Pursuit algorithm [74] by unrolling its greedy inference procedure into a fixed number of steps. At inference, the model starts with an initial estimate and residual (Algorithm 1, lines 3 and 2). At each iteration, MP-SAE greedily selects the feature that best aligns with the current residual by computing the inner product between the residual and each feature, and choosing the one with the highest projection (line 6). Once the best-matching feature is identified, the algorithm determines its contribution by projecting the residual onto that feature (line 7), adds this contribution to the current approximation of the input (line 8), and updates the residual accordingly (line 9). This procedure is repeated for T steps, producing a sparse code with $\|\boldsymbol{z}\|_0 \leq T$. The resulting encoding represents a sequential approximation of the input, constructed through greedy selection of locally optimal features that progressively refine the reconstruction of $\boldsymbol{x}$.

(*ii*) Conditional Orthogonality via Sequential Inference. A defining property of MP-SAE is that its greedy inference procedure promotes conditional orthogonality across selected features. This emerges from the residual update rule: once a concept $\boldsymbol{D}_{j^{(t-1)}}$ has been selected from the dictionary at iteration $t-1$, the updated residual $\boldsymbol{r}^{(t)}$ is orthogonal to it, as stated below formally.

Proposition 3.1 (Stepwise Orthogonality of MP Residuals)**.** *Let $\boldsymbol{r}^{(t)}$ be the residual at iteration t of MP-SAE inference, and let $\boldsymbol{D}_{j^{(t-1)}}$ be the feature selected at step $t-1$. Then:*

$$\boldsymbol{D}^\top_{j^{(t-1)}} \boldsymbol{r}^{(t)} = 0.$$

Each selected concept is removed from the residual subspace before the next selection, and subsequent concepts are chosen to explain what remains in the residual, orthogonal to the last selected feature. When trained, this yields a dictionary of features that are not globally orthogonal but are selectively chosen to be conditionally orthogonal. Although MP-SAE only enforces orthogonality to the most recently selected concept—unlike Orthogonal Matching Pursuit [89], which re-orthogonalizes against all prior selections—we observe empirically that residuals are often nearly orthogonal to all previously selected directions. This emergent behavior suggests the model implicitly promotes hierarchical separation in the learned dictionary, minimizing interference across levels. It aligns with the inductive bias of conditional orthogonality (Def. 2.3) and contrasts with standard SAEs, which tend to promote global quasi-orthogonality regardless of feature structure.

***(iii)* Access nonlinearly accessible features.** Beyond promoting conditional orthogonality, the residual-based inference structure of MP-SAE enables access to features that are nonlinearly embedded in the representation space. While each iteration applies a linear projection to the current residual, the residual itself evolves nonlinearly as a function of the input, due to its recursive dependence on previous selections. This results in a structured approximation of $\boldsymbol{x}$ that can be decomposed as:

$$\boldsymbol{x} = \boldsymbol{r}^{(0)} = \boldsymbol{\varphi}(\boldsymbol{x}) + \boldsymbol{r}^{(1)} = \underbrace{\boldsymbol{\varphi}(\boldsymbol{x})}_{\textit{linearly accessible}} + \underbrace{\sum_{t=1}^{T} \boldsymbol{\varphi}(\boldsymbol{r}^{(t)})}_{\textit{nonlinearly accessible}} + \underbrace{\boldsymbol{r}^{(T+1)}}_{\textit{residual error}},$$

where $\boldsymbol{\varphi}(\cdot)$ denotes the linear projection onto the selected feature at each step (See Fig. 3). The crucial insight is that, although each $\boldsymbol{\varphi}(\boldsymbol{r}^{(t)})$ is linear in its argument, the argument itself, $\boldsymbol{r}^{(t)}$, is a nonlinear function of $\boldsymbol{x}$. As a result, the composition $\boldsymbol{\varphi}(\boldsymbol{r}^{(t)})$ defines a feature that cannot be obtained from $\boldsymbol{x}$ via a single linear map. MP-SAE can thus potentially uncover higher-order concepts that are conditionally dependent on previously explained structure. This mechanism may be particularly relevant in settings where important features are entangled or nonlinearly composed such as hierarchies, temporal dependencies, or multimodal correlations. Moreover, it provides a constructive hypothesis for the phenomenon of "dark matter" in neural representations [71]: features that evade standard SAEs because they are not linearly *accessible* from the raw representation; that is, one can still have the LRH assumption of a linear mixing process hold, i.e., $x = Dz$, but other constraints start to relax (Def. 2.1). We return to this phenomenon in our empirical analysis.

4 Empirical Results

We now take a step towards uncovering challenges in SAEs emergent from assuming a partially correct model of neural network representations, i.e., LRH. Specifically, in Sec. 4.1, we analyze a synthetic domain that highlight the inability of existing SAEs to identify hierarchically structured concepts in a clearly defined domain. Building on these results, in Sec. 4.2, we analyze a natural vision–language domain to assess how features extracted under an inductive bias that accommodates hierarchical, nonlinearly accessible concepts—as in MP-SAE—differ from those identified by existing SAEs. In particular, we show that MP-SAE uniquely recovers cross-modal structure.

4.1 A Synthetic Generative Model of Hierarchical Features

We begin by evaluating whether SAEs, including our proposed MP-SAE, can recover conditionally orthogonal features using a synthetic hierarchy-based setup adapted from Bussmann et al. [20]. To formalize this setting, we introduce the following definition of a hierarchical generative process.

Definition 4.1 (Hierarchical Generative Process). *A generative process over* $\mathbf{p}$ *(parent concept) and* $\mathbf{c}$ *(child concept) is said to be* hierarchical *if their activations* z_p *and* z_c *satisfy*

$$\mathbb{P}(z_p > 0 \mid z_c > 0) = 1 \quad \textit{and} \quad \mathbb{P}(z_c > 0 \mid z_p > 0) < 1.$$

That is, a child's activation requires its parent's, whereas a parent may activate independently.

Based on this definition, we implemented the following hierarchical generative process. The ground truth $\boldsymbol{D}$ consists of 20 unit-norm concepts organized into a two-level tree: 11 disjoint parent concepts $\boldsymbol{D}_p$, 3 of which each have 3 disjoint children $\boldsymbol{D}_c$ as depicted in Fig. 2. Each input is generated by sampling a parent and, if present, a child, resulting in one or two active concepts.

$$\boldsymbol{x} = z_p \cdot \boldsymbol{D}_p + z_c \cdot \boldsymbol{D}_c$$

The activations z_p and z_c follow the activation pattern defined in Definition 4.1, taking values from $\mathcal{N}(1.5, {}^{1}\!/\!{}_{4^2})$ (ensuring positive values) when active and 0 otherwise.[3]

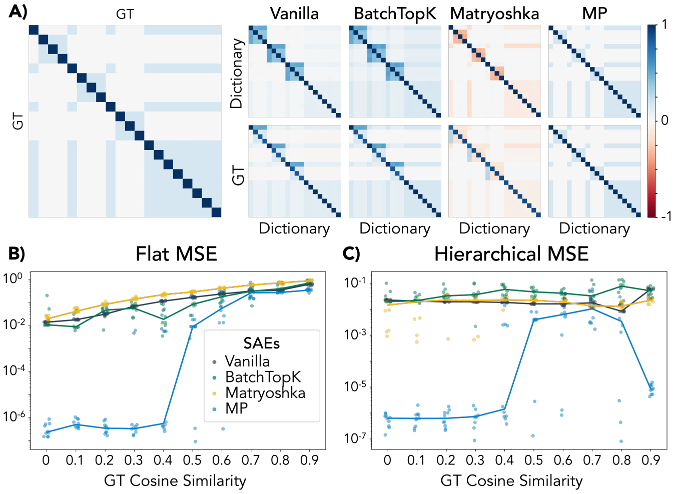


Figure 4: **Evaluating SAE on a hierarchical tree with controlled within-level similarity. A)** Correlation matrices for one similarity setting. Left shows the ground-truth matrix; the top row displays $\boldsymbol{D}^\top \boldsymbol{D}$ (self-similarity of learned features) and bottom row shows $\boldsymbol{D}_{\mathrm{GT}}^\top \boldsymbol{D}$ (alignment with ground truth). **Bottom:** Quantitative evaluation across varying levels of within-group correlation, median over 10 runs is reported. **B)** Flat MSE captures the deviation from the ground-truth intra-level correlation. **C)** Hierarchical MSE quantifies unintended correlations across levels.

We analyze Vanilla (ReLU) [23, 14], BatchTopK [17], and Matryoshka [20, 94] all trained with fixed ℓ_0 sparsity targets. In a fixed low intra-level interference regime (Fig. 4A), Vanilla and BatchTopK suffer from *feature absorption* [95], aligning child concepts with their parent and collapsing hierarchy—though they retain within-level corelation. Matryoshka avoids absorption and preserves hierarchy but introduces *negative interference* between siblings, distorting flat structure. Only MP-SAE recovers both intra and inter level structure. To further stress-test the different SAEs, we vary within-level correlation in the ground truth and evaluate learned dictionaries using "Flat MSE" (for intra-level alignment; see Fig. 4B) and "Hierarchical MSE" (for inter-level separation; see Fig. 4C). We observe that Matryoshka tends to preserves hierarchy but loses flat structure; meanwhile, Vanilla and BatchTopK behave oppositely. MP-SAE exhibits three regimes: low to moderate interference yields recovery of both structures; under high interference, both degrade, reaching eventually a point where we see MP-SAE sacrifice flatness to maintain hierarchy, highlighting its inductive bias.

Overall, this synthetic benchmark elicits challenges towards capturing hierarchical features via existing SAEs, including ones trained with explicit hierarchy-promoting objectives. Instead, capturing the relevant inductive bias via MP-SAE works really well: MP-SAE is uniquely capable of recovering hierarchically structured features when such property statistically exists. We now turn to pretrained representations to assess whether such structure arises in practice and how SAEs respond to it.

4.2 Representations from Pretrained Models

We analyze representations from pretrained vision-language models in the following experiments. This helps us avoid the challenge of flexibility of model representations in solely language-driven domains [73] (though see Sec. B.5 for preliminary results) Our evaluation is organized in four parts. We begin by (***i***) assessing MP-SAE's expressivity relative to existing SAEs. We then (***ii***) analyze the structure of its learned representations through effective rank and coherence metrics. Next, (***iii***) we test its robustness to inference-time sparsity variation. Finally, (***iv***) we investigate its ability to uncover multimodal features in vision-language models that remain inaccessible to existing SAEs.

Expressivity. We begin by evaluating the expressivity of different SAEs, including MP-SAE. Training was performed for 50 epochs with Adam, using a learning rate of $5 \cdot 10^{-4}$ and cosine decay to 10^{-6} with warmup. Models were trained on IN1K [96] train set, using frozen representations from the final layer of each backbone. For ViT-style models (e.g., DINOv2), all spatial tokens and the CLS token were included ($\sim$261 tokens per image for DinoV2, which results in approximately 25 billion training tokens for a training).

[3] We generalize the generative process from Bussmann et al. [20], where z_p and z_c are fixed across all inputs, making their firing magnitudes perfectly correlated ($z_p = \lambda z_c$). See Appendix B.1.4 for an extended discussion.

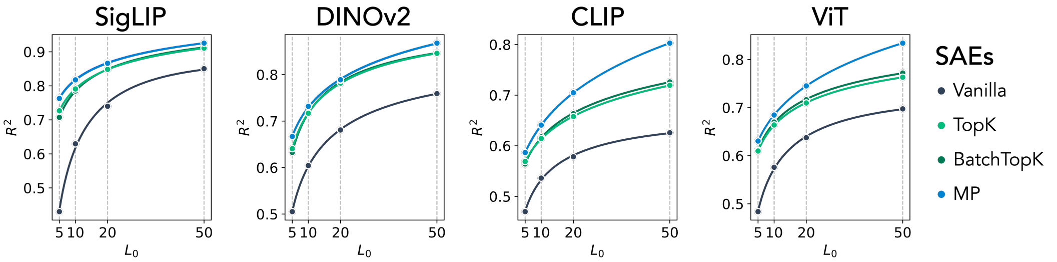


Figure 5: **MP-SAE recovers more expressive features than standard SAEs.** Reconstruction performance (R^2) as a function of sparsity level across four pretrained vision models: SigLIP, DINOv2, CLIP, and ViT. MP-SAE consistently achieves higher R^2 at comparable sparsity, indicating more efficient and informative decompositions.

Results are shown in Fig. 5, where we plot the Pareto frontier obtained by varying the sparsity level, using an expansion factor of 25 ($p = 25m$) for all SAEs. Across all tested models: SigLIP [97], DINOv2 [76], CLIP [75], and ViT [98], MP-SAE consistently achieves higher R^2 at comparable sparsity levels, indicating more efficient reconstruction. Our results above suggest that MP-SAE can explain a larger fraction of the representation space using fewer active features. Equivalently, its selected features are more informative per unit of sparsity. This aligns with the hypothesis that MP-SAE, through its iterative and residual-guided inference, can recover features that remain inaccessible to conventional SAEs. We note that, Engels et al. [71] identify a class of non-linearly accessible features that can't be recovered by linear sparse encoders. The improved expressivity observed here provides indirect evidence that MP-SAE may capture some of these otherwise hidden components. In the sections that follow, we examine more precisely the structure and semantics of the features recovered by MP-SAE to further probe this possibility.

Emergence of Rank Structure. To investigate how MP-SAE organizes features across varying sparsity levels, we analyze the effective rank of the co-activation matrix $\boldsymbol{Z}^\top \boldsymbol{Z}$, where $\boldsymbol{Z} \in \mathbb{R}^{n \times p}$ is the matrix of sparse codes across n inputs (see Fig. 6). Each entry of $\boldsymbol{Z}^\top \boldsymbol{Z}$ captures how frequently two concepts are co-activated. The effective rank, defined as the exponential of the entropy of the normalized eigenvalues of $\boldsymbol{Z}^\top \boldsymbol{Z}$ (Eq.2, Appendix B.3), measures the diversity of feature co-activation patterns across inputs. For standard SAEs, increasing the sparsity level k typically results in limited structural change: the encoder reuses similar subsets of features, leading to saturated rank and strong diagonal blocks in $\boldsymbol{Z}^\top \boldsymbol{Z}$. In contrast, MP-SAE shows a markedly different trend. As k increases, the effective rank grows steadily, reflecting a continual diversification of active feature sets. The rank growth induced by MP-SAE is thus **not an artifact of increased capacity, but a structural signal**: it reflects the model's ability to discover and disentangle latent modularity within representations.

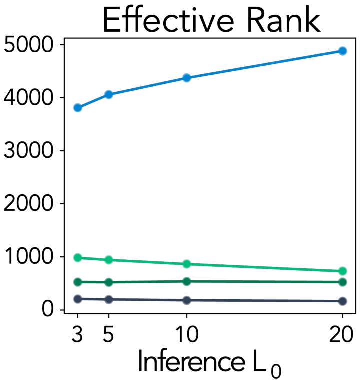


Figure 6: **Growth of effective rank as a function of sparsity k.** MP-SAE exhibits increasing combinatorial diversity, unlike standard SAEs whose co-activation structure quickly saturates.

Coherence and Conditional Orthogonality. We now examine the internal organization of the learned dictionaries by quantifying their coherence. Specifically, we use the Babel function [99], a standard metric in sparse approximation that captures cumulative interference between features. Given a dictionary $\boldsymbol{D} = [\boldsymbol{D}_1, \ldots, \boldsymbol{D}_p] \in \mathbb{R}^{m \times p}$ with unit-norm columns, the Babel function of order r is defined as:

$$\mu_1(r) = \max_{\substack{S \subset [p] \\ |S| = r}} \left(\max_{j \notin S} \sum_{i \in S} \left| \boldsymbol{D}_i^\top \boldsymbol{D}_j \right| \right)$$

Intuitively, $\mu_1(r)$ reflects how well a single concept can be approximated by a group of r others; lower values indicate better separability. Fig. 7 reports $\mu_1(r)$ for the learned dictionary, as well as the average over multiple representations for subsets of co-selected concepts at inference time. Interestingly, MP-SAE learns dictionaries with higher overall Babel scores than standard SAEs,

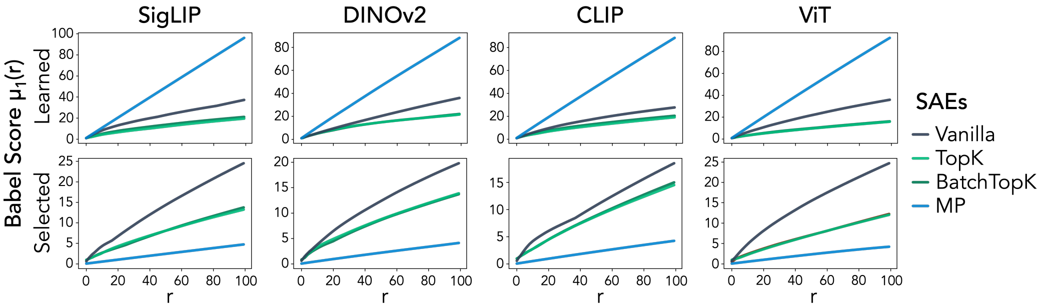


Figure 7: **MP-SAE promotes conditional orthogonality at inference.** Babel scores for full dictionaries (top) and co-activated subsets (bottom). MP-SAE dictionaries exhibit higher global coherence than standard SAEs, but select more separated features at inference.

indicating greater global interference. However, the concepts it selects during inference exhibit lower Babel scores, reflecting MP-SAE's tendency to construct conditionally orthogonal representations at inference (even when the full dictionary is correlated). By contrast, linear SAEs enforce global quasi-orthogonality in the dictionary but do not control which features co-activate at inference. As a result, inference often selects interfering directions despite a well-structured dictionary.

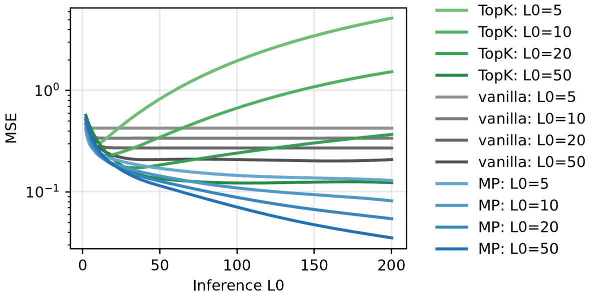


Figure 8: **Reconstruction error as a function of inference-time sparsity k.** When increasing SAEs' sparsity at inference on DINOv2 representations, we see MP-SAE exhibits monotonic improvement, while TopK SAEs may degrade due to sparsity mismatch. Similar results emerge in other settings (see Appendix B.5).

Adaptive Inference-Time Sparsity. A key property of MP-SAE is its ability to adaptively adjust the number of selected features k at inference time without retraining. Because inference proceeds via residual-based greedy selection, each additional step guarantees non-increasing reconstruction error. As shown in Fig. 8, MP-SAE is the only architecture for which reconstruction fidelity improves monotonically with k across all tested architectures and representation types. This stands in contrast to TopK SAEs, which degrade under sparsity mismatch: when trained with fixed k, the decoder implicitly specializes to superpositions of *exactly* k features, resulting in instability when k changes. ReLU-based SAEs, by contrast, cannot extend their support beyond what was active during training and exhibit flat or plateaued performance as k increases. This robustness is particularly salient given the epistemic uncertainty surrounding the "true sparsity" of neural representations. This property allows adjusting k at inference time without retraining, enabling controlled trade-offs between sparsity and reconstruction quality. Rather than fixing k, one can target a desired reconstruction level and let the model incrementally reveal the relevant features by adapting k.

Recovering Multimodal Concepts. We evaluate whether MP-SAE can recover shared concepts from vision-language models (VLMs). Our analysis focuses on representations from CLIP [75], with consistent results observed for AIMv2 [100], SigLIP [97], and SigLIP2[101] (See Appendix B.4). All models are evaluated on the COCO dataset [102], using both image and caption embeddings, balanced to ensure equal sampling across modalities. When trained on these joint embedding spaces, classical SAEs frequently learn *split dictionaries* [103, 104], where distinct features respond exclusively to either visual or textual inputs. This occurs despite the alignment of the underlying representation space and reflects a structural limitation in how these existing SAEs extract shared features. To quantify modality selectivity, we use the Modality Score from [105], defined for concept i as:

$$\text{ModalityScore}_i = \frac{\mathbb{E}_{\boldsymbol{z}\sim\iota}(z_i)}{\mathbb{E}_{\boldsymbol{z}\sim\iota}(z_i) + \mathbb{E}_{\boldsymbol{z}\sim\tau}(z_i)},$$

where ι and τ denote activations over image and text inputs, respectively. Scores near 1 indicate image specificity; near 0, text specificity; and intermediate values reflect balanced, multimodal activation.

Consistent with prior work [105], we find that standard SAEs yield sharply bimodal modality score distributions, confirming their tendency to separate modalities. In contrast, MP-SAE yields a significantly flatter distribution with substantial mass in the midrange (see Fig. 9), suggesting that it recovers genuinely multimodal units responsive to both modalities. This ability to extract multimodal concepts appears unique to MP-SAE. We hypothesize that MP-SAE's residual based inference plays a central role: once modality-specific information is explained, the residual shifts toward shared structure, allowing subsequent steps to capture cross-modal features (See Fig. 18). Prior work [104] has attempted to bridge the modality gap by applying a learned translation from one modality to the other. This can be seen as equivalent to a single inference step in MP-SAE. However, MP-SAE continues this process iteratively, progressively refining the residual and revealing joint semantic structure. This iterative mechanism enables MP-SAE to extract hierarchical and non-linearly accessible multimodal features beyond the scope of conventional SAEs.

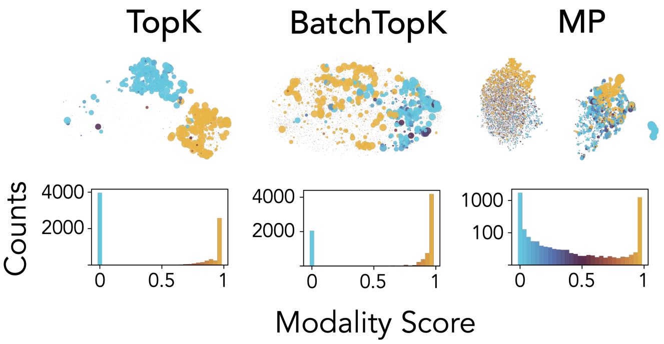


Figure 9: **MP-SAE identifies shared structures across modalities. Top:** UMAP of sparse codes (yellow denotes image representations; blue denotes text). **Bottom:** Distribution of modality scores. We see existing SAEs yield features skewed towards a specific modality, suggesting bimodal structure in model representations. However, MP-SAE uniquely recovers multimodal units with shared activation across text and vision inputs.

5 Conclusion

In this work, we revisit SAEs to understand their ability to identify concepts encoded in a manner outside the scope of their motivating hypothesis, i.e., LRH. Our results show that while standard SAEs are indeed effective under LRH, they struggle to capture more complex representational structures—such as hierarchical, nonlinearly accessible, and multimodal features—that have emerged in recent studies of large neural networks. To contextualize these results with respect to an SAE that is motivated to (partially) accommodate such phenomenology outside LRH's scope, we introduced MP-SAE. Specifically, an extension of the classical Matching Pursuit algorithm, MP-SAE promotes conditional orthogonality through residual-guided inference. Our results show that this design enables MP-SAE to recover rich, hierarchically organized, and nonlinearly accessible features that standard SAEs missed. Beyond interpretability, we also find that residual-based, sequential encoders like MP-SAE offer practical advantages, such as adaptive inference-time sparsity and progressive feature discovery, which may be of independent interest to the community. *Overall, we argue our results highlight the limitations of assuming purely linear structure in neural representations.*

Limitations We emphasize that our goal with proposing MP-SAE was not to propose a "superior" SAE architecture, but to use it as a tool whose inductive bias aligns with a particular class of concept structures, namely, hierarchically and nonlinearly accessible features. Under this hypothesis, MP-SAE proves more expressive, more modular, and more robust than standard SAEs. However, we note MP-SAE assumes that representations can be incrementally explained by conditionally orthogonal features. This inductive bias may be poorly matched in flat or entangled regimes, or in settings with categorical structure. Matching Pursuit is also a greedy algorithm, which lacks global optimality and may be brittle under extreme noise (though we do find existing SAEs fail in this regime as well).

Acknowledgements

Authors thank the CRISP Group at Harvard SEAS for insightful conversations, and the Kempner Institute and CBS-NTT program in Physics of Intelligence at Harvard University for access to compute resources used for performing experiments reported in this paper. Valérie Costa thanks the Bertarelli Foundation for supporting her work as a fellow.

References

[1] OpenAI. Gpt-4 technical report, 2024.

[2] OpenAI. Openai o3 and o4-mini system card, 2025.

[3] Gemini Team. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[4] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[5] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[6] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[7] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[8] OpenAI. Sora: Creating video from text, 2024.

[9] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[10] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *arXiv preprint arXiv:2409.17146*, 2024.

[11] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *preprint arXiv:1702.08608*, 2017.

[12] Usman Anwar, Abulhair Saparov, Javier Rando, Daniel Paleka, Miles Turpin, Peter Hase, Ekdeep Singh Lubana, Erik Jenner, Stephen Casper, Oliver Sourbut, et al. Foundational challenges in assuring alignment and safety of large language models. *ArXiv e-print*, 2024.

[13] Paolo Tripicchio and Salvatore D'Avella. Is deep learning ready to satisfy industry needs? *Procedia Manufacturing*, 2020.

[14] 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.

[15] Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

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

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

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

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

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

[22] Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024.

[23] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse Autoencoders Find Highly Interpretable Features in Language Models, October 2023. arXiv:2309.08600 [cs].

[24] 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.

[25] Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*, 2023.

[26] Kaarel Hänni, Jake Mendel, Dmitry Vaintrob, and Lawrence Chan. Mathematical models of computation in superposition. *arXiv preprint arXiv:2408.05451*, 2024.

[27] Micah Adler, Dan Alistarh, and Nir Shavit. Towards combinatorial interpretability of neural computation. *arXiv preprint arXiv:2504.08842*, 2025.

[28] Micah Adler and Nir Shavit. On the complexity of neural computation in superposition. *arXiv preprint arXiv:2409.15318*, 2024.

[29] William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In Richard Beals, Anatole Beck, Alexandra Bellow, and Arshag Hajian, editors, *Contemporary Mathematics*, volume 26, pages 189–206. American Mathematical Society, Providence, Rhode Island, 1984.

[30] Kasper Green Larsen and Jelani Nelson. The johnson-lindenstrauss lemma is optimal for linear dimensionality reduction. *arXiv preprint arXiv:1411.2404*, 2014.

[31] Michael Elad. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer Science & Business Media, August 2010. Google-Books-ID: d5b6lJI9BvAC.

[32] Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

[33] Can Demircan, Tankred Saanum, Akshay K Jagadish, Marcel Binz, and Eric Schulz. Sparse autoencoders reveal temporal difference learning in large language models. *arXiv preprint arXiv:2410.01280*, 2024.

[34] Yida Chen, Fernanda Viégas, and Martin Wattenberg. Beyond surface statistics: Scene representations in a latent diffusion model. *arXiv preprint arXiv:2306.05720*, 2023.

[35] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B Tenenbaum, William T Freeman, and Antonio Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. *arXiv preprint arXiv:1811.10597*, 2018.

[36] David Bau, Steven Liu, Tongzhou Wang, Jun-Yan Zhu, and Antonio Torralba. Rewriting a deep generative model. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 351–369. Springer, 2020.

[37] Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. Labels4free: Unsupervised segmentation using stylegan. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13970–13979, 2021.

[38] Xiaodan Du, Nicholas Kolkin, Greg Shakhnarovich, and Anand Bhattad. Generative models: What do they know? do they know things? let's find out! *arXiv preprint arXiv:2311.17137*, 2023.

[39] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *International conference on machine learning*, pages 9786–9796. PMLR, 2020.

[40] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9243–9252, 2020.

[41] Yossi Gandelsman, Alexei A Efros, and Jacob Steinhardt. Interpreting clip's image representation via text-based decomposition. *arXiv preprint arXiv:2310.05916*, 2023.

[42] Jack Merullo, Louis Castricato, Carsten Eickhoff, and Ellie Pavlick. Linearly mapping from image to text space. *arXiv preprint arXiv:2209.15162*, 2022.

[43] Matthew Kowal, Achal Dave, Rares Ambrus, Adrien Gaidon, Konstantinos G Derpanis, and Pavel Tokmakov. Understanding video transformers via universal concept discovery. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[44] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom In: An Introduction to Circuits. *Distill*, 5(3):e00024.001, March 2020.

[45] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017.

[46] Chris Olah, Nick Cammarata, Chelsea Voss, Ludwig Schubert, and Gabriel Goh. Naturally occurring equivariance in neural networks. *Distill*, 5(12):e00024–004, 2020.

[47] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. An overview of early vision in inceptionv1. *Distill*, 5(4):e00024–002, 2020.

[48] Nick Cammarata, Gabriel Goh, Shan Carter, Ludwig Schubert, Michael Petrov, and Chris Olah. Curve detectors. *Distill*, 5(6):e00024–003, 2020.

[49] Ludwig Schubert, Chelsea Voss, Nick Cammarata, Gabriel Goh, and Chris Olah. High-low frequency detectors. *Distill*, 6(1):e00024–005, 2021.

[50] Thomas Fel, Louis Bethune, Andrew Lampinen, Thomas Serre, and Katherine Hermann. Understanding visual feature reliance through the lens of complexity. *Advances in Neural Information Processing Systems*, 37:69888–69924, 2024.

[51] Thomas Fel, Victor Boutin, Mazda Moayeri, Rémi Cadène, Louis Bethune, Léo andéol, Mathieu Chalvidal, and Thomas Serre. A Holistic Approach to Unifying Automatic Concept Extraction and Concept Importance Estimation, October 2023. arXiv:2306.07304 [cs].

[52] Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36:41451–41530, 2023.

[53] Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*, 2023.

[54] Andrew Lee, Xiaoyan Bai, Itamar Pres, Martin Wattenberg, Jonathan K Kummerfeld, and Rada Mihalcea. A mechanistic understanding of alignment algorithms: A case study on dpo and toxicity. *arXiv preprint arXiv:2401.01967*, 2024.

[55] Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. Linear representations of sentiment in large language models. *arXiv preprint arXiv:2310.15154*, 2023.

[56] Niru Maheswaranathan, Alex Williams, Matthew Golub, Surya Ganguli, and David Sussillo. Reverse engineering recurrent networks for sentiment classification reveals line attractor dynamics. *Advances in neural information processing systems*, 32, 2019.

[57] Niru Maheswaranathan, Alex H Williams, Matthew D Golub, Surya Ganguli, and David Sussillo. Line attractor dynamics in recurrent networks for sentiment classification. In *ICML 2019 Workshop on Identifying and Understanding Deep Learning Phenomena*, 2019.

[58] Wentao Zhu, Zhining Zhang, and Yizhou Wang. Language models represent beliefs of self and others. *arXiv preprint arXiv:2402.18496*, 2024.

[59] Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent Linear Representations in World Models of Self-Supervised Sequence Models, September 2023. arXiv:2309.00941 [cs].

[60] Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. Language models implement simple word2vec-style vector arithmetic. *arXiv preprint arXiv:2305.16130*, 2023.

[61] Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *arXiv preprint arXiv:2406.11717*, 2024.

[62] Samyak Jain, Ekdeep S Lubana, Kemal Oksuz, Tom Joy, Philip Torr, Amartya Sanyal, and Puneet Dokania. What makes and breaks safety fine-tuning? a mechanistic study. *Advances in Neural Information Processing Systems*, 37:93406–93478, 2024.

[63] Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. Linearity of relation decoding in transformer language models. *arXiv preprint arXiv:2308.09124*, 2023.

[64] Jack Merullo, Noah A Smith, Sarah Wiegreffe, and Yanai Elazar. On linear representations and pretraining data frequency in language models. *arXiv preprint arXiv:2504.12459*, 2025.

[65] Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. A Latent Variable Model Approach to PMI-based Word Embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399, 2016. Place: Cambridge, MA Publisher: MIT Press.

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

[67] Martin Wattenberg and Fernanda B. Viégas. Relational Composition in Neural Networks: A Survey and Call to Action, July 2024. arXiv:2407.14662 [cs].

[68] Lewis Smith. The 'strong' feature hypothesis could be wrong, August 2024.

[69] Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. The geometry of categorical and hierarchical concepts in large language models. *arXiv preprint arXiv:2406.01506*, 2024.

[70] Róbert Csordás, Christopher Potts, Christopher D. Manning, and Atticus Geiger. Recurrent Neural Networks Learn to Store and Generate Sequences using Non-Linear Representations, August 2024. arXiv:2408.10920 [cs].

[71] Joshua Engels, Logan Riggs, and Max Tegmark. Decomposing The Dark Matter of Sparse Autoencoders, March 2025. arXiv:2410.14670 [cs].

[72] Joshua Engels, Eric J. Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. Not All Language Model Features Are One-Dimensionally Linear, February 2025. arXiv:2405.14860 [cs].

[73] Core Francisco Park, Andrew Lee, Ekdeep Singh Lubana, Yongyi Yang, Maya Okawa, Kento Nishi, Martin Wattenberg, and Hidenori Tanaka. ICLR: In-Context Learning of Representations, May 2025. arXiv:2501.00070 [cs].

[74] S.G. Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

[75] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[76] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[77] Yibo Jiang, Goutham Rajendran, Pradeep Ravikumar, Bryon Aragam, and Victor Veitch. On the origins of linear representations in large language models. *arXiv preprint arXiv:2403.03867*, 2024.

[78] Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

[79] Zihao Wang, Lin Gui, Jeffrey Negrea, and Victor Veitch. Concept Algebra for (Score-Based) Text-Controlled Generative Models, February 2024. arXiv:2302.03693 [cs].

[80] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic Regularities in Continuous Space Word Representations. In Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff, editors, *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

[81] 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*, pages 689–696, 2009.

[82] Alireza Makhzani and Brendan Frey. k-sparse autoencoders, 2014.

[83] Andrew M Saxe, James L McClelland, and Surya Ganguli. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116(23):11537–11546, 2019.

[84] Matthew Lyle Olson, Musashi Hinck, Neale Ratzlaff, Changbai Li, Phillip Howard, Vasudev Lal, and Shao-Yen Tseng. Analyzing hierarchical structure in vision models with sparse autoencoders. *arXiv preprint arXiv:2505.15970*, 2025.

[85] Maya Okawa, Ekdeep S Lubana, Robert Dick, and Hidenori Tanaka. Compositional abilities emerge multiplicatively: Exploring diffusion models on a synthetic task. *Advances in Neural Information Processing Systems*, 36:50173–50195, 2023.

[86] Ekdeep Singh Lubana, Kyogo Kawaguchi, Robert P Dick, and Hidenori Tanaka. A percolation model of emergence: Analyzing transformers trained on a formal language. *arXiv preprint arXiv:2408.12578*, 2024.

[87] Tian Qin, Naomi Saphra, and David Alvarez-Melis. Sometimes i am a tree: Data drives unstable hierarchical generalization. *arXiv preprint arXiv:2412.04619*, 2024.

[88] Angelica Chen, Ravid Shwartz-Ziv, Kyunghyun Cho, Matthew L Leavitt, and Naomi Saphra. Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in mlms. *arXiv preprint arXiv:2309.07311*, 2023.

[89] Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, pages 40–44 vol.1, 1993.

[90] Scott Shaobing Chen, David L Donoho, and Michael A Saunders. Atomic decomposition by basis pursuit. *SIAM review*, 43(1):129–159, 2001.

[91] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457, 2004.

[92] Thomas Blumensath and Mike E Davies. Iterative thresholding for sparse approximations. *Journal of Fourier analysis and Applications*, 14(5-6):629–654, 2008.

[93] Lorenzo Peotta and Pierre Vandergheynst. Matching pursuit with block incoherent dictionaries. *IEEE transactions on signal processing*, 55(9):4549–4557, 2007.

[94] Vladimir Zaigrajew, Hubert Baniecki, and Przemyslaw Biecek. Interpreting CLIP with Hierarchical Sparse Autoencoders, February 2025. arXiv:2502.20578 [cs].

[95] David Chanin, James Wilken-Smith, Tomas Dulka, Hardik Bhatnagar, and Joseph Bloom. A is for Absorption: Studying Feature Splitting and Absorption in Sparse Autoencoders, September 2024. arXiv:2409.14507 [cs].

[96] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[97] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023.

[98] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[99] J.A. Tropp. Greed is good: algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, October 2004.

[100] Enrico Fini, Mustafa Shukor, Xiujun Li, Philipp Dufter, Michal Klein, David Haldimann, Sai Aitharaju, Victor Guilherme Turrisi da Costa, Louis Béthune, Zhe Gan, et al. Multimodal autoregressive pre-training of large vision encoders. *ArXiv e-print*, 2024.

[101] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *ArXiv e-print*, 2025.

[102] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pages 740–755. Springer, 2014.

[103] Jayneel Parekh, Pegah Khayatan, Mustafa Shukor, Alasdair Newson, and Matthieu Cord. A concept-based explainability framework for large multimodal models. *ArXiv e-print*, 2024.

[104] Usha Bhalla, Suraj Srinivas, Asma Ghandeharioun, and Himabindu Lakkaraju. Towards unifying interpretability and control: Evaluation via intervention. *ArXiv e-print*, 2024.

[105] Isabel Papadimitriou, Huangyuan Su, Thomas Fel, Naomi Saphra, Sham Kakade, and Stephanie Gil. Interpreting the linear structure of vision-language model embedding spaces. *arXiv preprint arXiv:2504.11695*, 2025.

[106] 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.

[107] Michael Lustig, David Donoho, and John M. Pauly. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/mrm.21391.

[108] Keijo Hämäläinen, Aki Kallonen, Ville Kolehmainen, Matti Lassas, Kati Niinimäki, and Samuli Siltanen. Sparse Tomography. *SIAM Journal on Scientific Computing*, 35(3):B644–B665, January 2013. Publisher: Society for Industrial and Applied Mathematics.

[109] Leyuan Fang, Shutao Li, Ryan P. McNabb, Qing Nie, Anthony N. Kuo, Cynthia A. Toth, Joseph A. Izatt, and Sina Farsiu. Fast Acquisition and Reconstruction of Optical Coherence Tomography Images via Sparse Representation. *IEEE Transactions on Medical Imaging*, 32(11):2034–2049, November 2013.

[110] Julien Mairal, Michael Elad, and Guillermo Sapiro. Sparse representation for color image restoration. *IEEE Transactions on image processing*, 17(1):53–69, 2007.

[111] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Non-local sparse models for image restoration. In *2009 IEEE 12th international conference on computer vision*, pages 2272–2279. IEEE, 2009.

[112] Weisheng Dong, Lei Zhang, Guangming Shi, and Xin Li. Nonlocally Centralized Sparse Representation for Image Restoration. *IEEE Transactions on Image Processing*, 22(4):1620–1630, April 2013.

[113] Lee C Potter, Emre Ertin, Jason T Parker, and Müjdat Cetin. Sparsity and compressed sensing in radar imaging. *Proceedings of the IEEE*, 98(6):1006–1020, 2010.

[114] Brian Cleary, Le Cong, Anthea Cheung, Eric S. Lander, and Aviv Regev. Efficient generation of transcriptomic profiles by random composite measurements. *Cell*, 171(6):1424–1436.e18, 2017.

[115] Brian Cleary, Brooke Simonton, Jon Bezney, Evan Murray, Shahul Alam, Anubhav Sinha, Ehsan Habibi, Jamie Marshall, Eric S Lander, Fei Chen, et al. Compressed sensing for highly efficient imaging transcriptomics. *Nature Biotechnology*, pages 1–7, 2021.

[116] Joseph Lucas, Carlos Carvalho, Quanli Wang, Andrea Bild, Joseph R. Nevins, and Mike West. Sparse Statistical Modelling in Gene Expression Genomics. In Kim-Anh Do, Marina Vannucci, and Peter Müller, editors, *Bayesian Inference for Gene Expression and Proteomics*, pages 155–176. Cambridge University Press, Cambridge, 2006.

[117] Daniela M. Witten and Robert J. Tibshirani. Extensions of sparse canonical correlation analysis with applications to genomic data. *Statistical Applications in Genetics and Molecular Biology*, 8(1):Article28, 2009.

[118] B. K. Natarajan. Sparse Approximate Solutions to Linear Systems. *SIAM Journal on Computing*, 24(2):227–234, April 1995. Publisher: Society for Industrial and Applied Mathematics.

[119] Elaine Crespo Marques, Nilson Maciel, Lírida Naviner, Hao Cai, and Jun Yang. A Review of Sparse Recovery Algorithms. *IEEE Access*, 7:1300–1322, 2019.

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

[121] Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2):489–509, 2006.

[122] Emmanuel J Candès and Michael B Wakin. An introduction to compressive sampling. *IEEE signal processing magazine*, 25(2):21–30, 2008.

[123] Miles Lopes. Estimating unknown sparsity in compressed sensing. In *International Conference on Machine Learning*, pages 217–225. PMLR, 2013.

[124] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.

[125] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[126] J.A. Tropp. Just relax: convex programming methods for identifying sparse signals in noise. *IEEE Transactions on Information Theory*, 52(3):1030–1051, March 2006.

[127] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical learning with sparsity: the lasso and generalizations*. CRC press, 2015.

[128] Thomas Blumensath and Mike E Davies. Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis*, 27(3):265–274, 2009.

[129] Ron Rubinstein, Alfred M. Bruckstein, and Michael Elad. Dictionaries for Sparse Representation Modeling. *Proceedings of the IEEE*, 98(6):1045–1057, June 2010.

[130] Ivana Tosic and Pascal Frossard. Dictionary Learning. *IEEE Signal Processing Magazine*, 28(2):27–38, March 2011.

[131] K. Engan, S.O. Aase, and J. Hakon Husoy. Method of optimal directions for frame design. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 2443–2446 vol.5, 1999.

[132] Niladri S Chatterji and Peter L Bartlett. Alternating minimization for dictionary learning: Local convergence guarantees. *arXiv:1711.03634*, pages 1–26, 2017.

[133] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[134] Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 399–406, Madison, WI, USA, 2010. Omnipress.

[135] Xiaohan Chen, Jialin Liu, Zhangyang Wang, and Wotao Yin. Theoretical linear convergence of unfolded ista and its practical weights and thresholds. In *Proceedings of Advances in Neural Information Processing Systems*, volume 31, pages 1–11, 2018.

[136] Pierre Ablin, Thomas Moreau, Mathurin Massias, and Alexandre Gramfort. Learning step sizes for unfolded sparse coding. In *Proceedings of Advances in Neural Information Processing Systems*, volume 32, pages 1–11, 2019.

[137] Bahareh Tolooshams and Demba E Ba. Stable and interpretable unrolled dictionary learning. *Transactions on Machine Learning Research*, 2022.

[138] Benoît Malézieux, Thomas Moreau, and Matthieu Kowalski. Understanding approximate and unrolled dictionary learning for pattern recovery. In *International Conference on Learning Representations*, 2022.

[139] Akshay Rangamani, Anirbit Mukherjee, Amitabh Basu, Ashish Arora, Tejaswini Ganapathi, Sang Chin, and Trac D. Tran. Sparse coding and autoencoders. In *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, pages 36–40, 2018.

[140] Thanh V Nguyen, Raymond KW Wong, and Chinmay Hegde. On the dynamics of gradient descent for autoencoders. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, pages 2858–2867. PMLR, 2019.

[141] Sanjeev Arora, Rong Ge, Tengyu Ma, and Ankur Moitra. Simple, efficient, and neural algorithms for sparse coding. In Peter Grünwald, Elad Hazan, and Satyen Kale, editors, *Proceedings of Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pages 113–149, Paris, France, 03–06 Jul 2015. PMLR.

[142] Thomas Fel, Agustin Picard, Louis Bethune, Thibaut Boissin, David Vigouroux, Julien Colin, Rémi Cadène, and Thomas Serre. CRAFT: Concept Recursive Activation FacTorization for Explainability, March 2023. arXiv:2211.10154 [cs].

[143] Kola Ayonrinde, Michael T Pearce, and Lee Sharkey. Interpretability as compression: Reconsidering sae explanations of neural activations with mdl-saes. *arXiv preprint arXiv:2410.11179*, 2024.

[144] Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European signal processing conference*, pages 606–610. IEEE, 2007.

[145] Ronen Eldan. HuggingFace: TinyStories-1M Models, 2023.

[146] Ronen Eldan and Yuanzhi Li. Tinystories: How small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*, 2023.

[147] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Appendix

Table of Contents

A Related Work

Sparse coding Sparse linear generative models aim to find a sparse representations $\boldsymbol{z} \in \mathbb{R}^p$ that explains the data $\boldsymbol{x} \in \mathbb{R}^m$ via a collection of atoms, forming an overcomplete dictionary $\boldsymbol{D} \in \mathbb{R}^{m \times p}$ $(p \gg m.)$ [106, 32]. Sparse representations are ubiquitous in science and engineering, with origins from computational neuroscience [106, 32] and applications in medical imaging [107–109], image restorations [110–112], radar sensing [113], transcriptomics [114, 115], and genomics [116, 117].

When the dictionary is fixed, finding the sparsest $\boldsymbol{z}$ decomposition of the input $\boldsymbol{x}$ is known as a sparse approximation problem (i.e., minimizing the ℓ_0 norm of the representation) [118, 99, 119]. This combinatorial sparse problem is NP-hard and has been the focus of extensive research in compressed sensing [120–123] and signal processing [31, 124]. Classical approaches include greedy ℓ_0-based algorithms such as Matching Pursuit [74] and Orthogonal Matching Pursuit [89], as well as the convex relaxation ℓ_1-based methods [90, 125–127] such as Iterative Soft-Thresholding Algorithm (ISTA) [91, 92] and Iterative Hard-Thresholding (IHT) [128]. Since sparse recovery lacks a closed-form solution [118], the sparse approximation step typically requires multiple residual-based iterations to converge.

When the dictionary is learned jointly with the sparse codes, the problem is referred to as sparse coding [32] or dictionary learning [129, 31, 130]. Classical approaches such as MOD [131] and K-SVD [124] solve the problem using alternating minimization [132]. The bottleneck of sparse coding is in the convergence rate of the inner problem, which relates to the properties on dictionary atoms; while the slow sublinear convergence of inner iterative algorithms such as ISTA [91, 92] can be accelerated via optimization techniques (e.g., adding momentum [133], there exist no optimization algorithm that can guarantee linear convergence for general sparse coding problem.

Unrolling for sparse coding To address the slow convergence of sparse coding, in 2010, learned ISTA (LISTA) [134] sparked a new approach, which is now known as unrolling, to turn and relax the iterations of ISTA into layers of a neural network to accelerate the inner problem. Since then, numerous works have theoretically studied the linear convergence of unrolled ISTA [135, 136], and some others have highlighted the implicit acceleration that this approach may bring in learning the dictionary [137, 138]. Moreover, a parallel literature has theoretically shown that approximate sparse coding, such as a shallow ReLU network [139, 140], or shallow hard-thresholding (recently named as JumpReLU [18]) [141, 140] is practically enough to perform dictionary learning.

Interpretability of Neural Networks Recently, [51] has shown that many interpretability methods used to extract concepts were in fact sparse coding methods. The field has re-emerged as a compelling framework for interpreting the internal mechanisms of high-performing, large-scale models [142, 1–10], particularly in light of persistent challenges surrounding safety and interpretability [11–13]. Within this renewed interest, Sparse Autoencoders (SAEs) have gained prominence as a principled method for exposing the latent structure of neural representations [14–23].

A significant portion of recent interpretability research is underpinned by the Linear Representation Hypothesis (LRH) [24, 25, 25], which posits that high-dimensional neural representations can be decomposed as superpositions over a large set of approximately orthogonal directions, each aligned with human-interpretable concepts. This hypothesis has been theoretically substantiated through formal complexity bounds and constructive frameworks for linearly decodable computation in superposition [26–28].

Empirical support for LRH spans both vision and language domains. In vision models, internal representations have been shown to encode factors such as geometry, lighting, facial structure, and scene composition [34–50]. In language models, analogous patterns emerge, with latent directions capturing syntactic and semantic content [52–57], character roles, arithmetic concepts [58–60], relational structures [63–66], and even safety-related behaviors such as refusal to generate harmful outputs [61, 62].

However, recent work has challenged the universality of LRH, arguing that the geometric assumptions behind SAEs may not hold in practice [67, 68]. Several studies reveal nonlinear and non-directional structures: hierarchical concepts may align with orthogonal axes, but categorical ones form overlapping polytopes [69]; RNNs exhibit “onion-like” layers [70]; and temporal concepts like weekdays often require multi-dimensional, context-sensitive representations [71–73, 21]. These findings highlight the need for improved SAE architectures to capture interpretable concepts that lie beyond the assumptions of the LRH.

B Extended Experimental Details and Further Results

B.1 Synthetic Experiment

The synthetic experiments were conducted using the codebase provided by Bussmann et al. [20]. Specifically, we introduced the following modifications:

- Extended the framework to support both MP-SAE and BatchTopK variants.
- Enforced mutual exclusivity among child features, yielding a sparsity of 1 or 2.
- Introduced variance in the firing magnitudes of the features (see Section B.1.4).
- Added controlled intra-level correlations between features.

Before detailing our experimental setup, we first introduce the concept of Matryoshka Sparse Autoencoders (SAEs).

Matryoshka Sparse Autoencoders (SAEs) Matryoshka SAEs [20, 94] extend traditional sparse autoencoders by enforcing accurate reconstruction from multiple nested prefixes of the latent vector. Given an input $\boldsymbol{x} \in \mathbb{R}^m$, the encoder computes $\boldsymbol{z} = \boldsymbol{\Pi}\{\boldsymbol{W}^\top(\boldsymbol{x} - \boldsymbol{b}_{\text{pre}}) + \boldsymbol{b}\} \in \mathbb{R}^p$, and each prefix $m_i \in \mathcal{M}$ is used to reconstruct $\boldsymbol{x}$ via the first m_i latents:

$$\hat{\boldsymbol{x}}^{(m_i)} = \boldsymbol{D}_{0:m_i,:}\boldsymbol{z}_{0:m_i} + \boldsymbol{b}_{\text{pre}}.$$

The set $\mathcal{M} = m_1, m_2, \ldots, m_n$ defines a hierarchy of prefix lengths such that $m_1 < m_2 < \cdots < m_n = p$, where each m_i corresponds to a sub-SAE tasked with reconstructing the input from a progressively longer portion of the latent code. In practice, $\mathcal{M}$ can be fixed before training or sampled stochastically per batch to encourage robustness across multiple levels of abstraction.

The total loss encourages reconstructions at multiple levels:

$$\mathcal{L}(\boldsymbol{x}) = \sum_{m_i \in \mathcal{M}} \|\boldsymbol{x} - \hat{\boldsymbol{x}}^{(m_i)}\|_2^2 + \alpha\mathcal{L}_{\text{aux}}.$$

This design encourages early latents to encode general features and later ones to specialize, improving disentanglement and reducing feature absorption [95] in hierarchical settings.

B.1.1 Dataset Generation

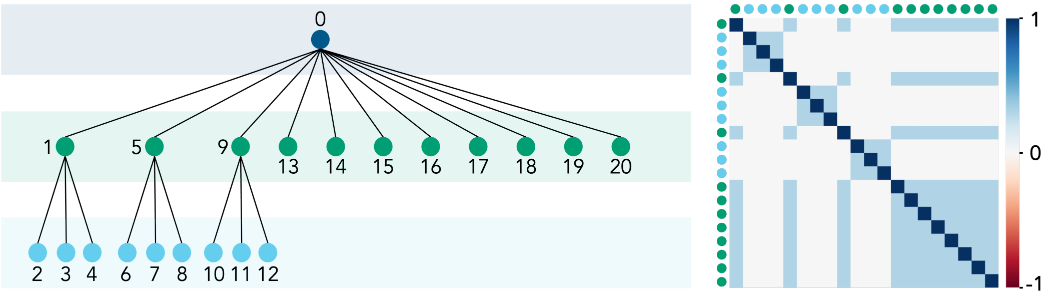


Figure 10: Hierarchical tree structure (from [20], also shown in Figure 4) with node indices. Indices are omitted in subsequent figures to avoid visual clutter.

The tree structure contains 20 nodes, each corresponding to a unit-norm vector in $\mathbb{R}^{20}$, as illustrated in Figure 2 and detailed in Figure 10. The exact tree configuration used in our experiments is available on the project's GitHub repository[4].

Feature Activations The root node (index 0) is fixed to the zero vector, i.e., $0 \in \mathbb{R}^{20}$. All child nodes are mutually exclusive, leading to input sparsity levels of either 1 or 2. Leaf nodes—green parents without children (indices 13–20)—are independently activated with probability 0.05 ($\mathbb{P}[z_p >$

[4] https://github.com/mpsae/MP-SAE

$0] = 0.05$). Internal green parent nodes (1, 5, and 9) are activated with probability 0.2 ($\mathbb{P}[z_p > 0] = 0.2$). Blue child nodes (2–4, 6–8, 10–12) are activated with probability 0.2 ($\mathbb{P}[z_c > 0 \mid z_p > 0] = 0.2$), conditional on their corresponding parent (1, 5, or 9) being active, in line with Definition 4.1. This results in an expected target sparsity of $\ell_0 = 1.36$.

Activation Magnitudes If a node is active, its sparse code is drawn from a Gaussian distribution $z \sim \mathcal{N}(1.5, {}^1\!/_{4^2})$. This differs from the generative process of Bussmann et al. [20], where z_p and z_c are fixed across all inputs, making their firing magnitudes perfectly correlated ($z_p = \lambda z_c$). We provide a justification for this design choice in Section B.1.4. The input $\boldsymbol{x}$ is then constructed as the sum of the active node vectors scaled by their corresponding code values:

$$\boldsymbol{x} = z_p \cdot \boldsymbol{D}_p + z_c \cdot \boldsymbol{D}_c.$$

Introducing Intra-Level Correlations. In the original codebase, all dictionary directions are orthogonal, obtained via QR decomposition. To introduce correlations specifically between nodes at the same hierarchical level, we perturb each dictionary element as

$$\boldsymbol{D}_i \leftarrow (1 - \varepsilon)\boldsymbol{D}_i + \varepsilon \sum_{j \neq i} \boldsymbol{D}_j,$$

where the sum is restricted to elements sharing the same parent, followed by renormalization. The parameter ε is tuned empirically to achieve the desired degree of intra-level correlation.

B.1.2 Training

Our training pipeline follows the same procedure as that used by Bussmann et al. [20]. Samples are drawn from the tree structure in batches of 200 over 15,000 training steps, using the Adam optimizer with $\beta = (0.5, 0.9375)$ and a learning rate of 3×10^{-2}. To stabilize optimization, gradient norms are clipped at 1.

For sparsity control, all SAEs, except MP-SAE, are trained to match a target ℓ_0 sparsity of 1.36.

- Vanilla and Matryoshka SAEs begin with a low ℓ_1 regularization during the first 3,000 steps, which is subsequently increased or decreased based on whether the observed sparsity is below or above the target.
- BatchTopK SAE is trained with a fixed sparsity of 3 for the initial 1,000 steps, after which the sparsity level is reduced to 1.36 for the remainder of training.
- MP-SAE, in contrast, does not need to rely on a fixed sparsity target. (make it sound more like an advantage)Instead, it unrolls the encoder until either the residual norm drops below 0.05 or the support of the sparse code z stabilizes—avoiding infinite loops when the residual cannot be sufficiently reduced. Unlike the other SAEs, MP-SAE uses tied weights.

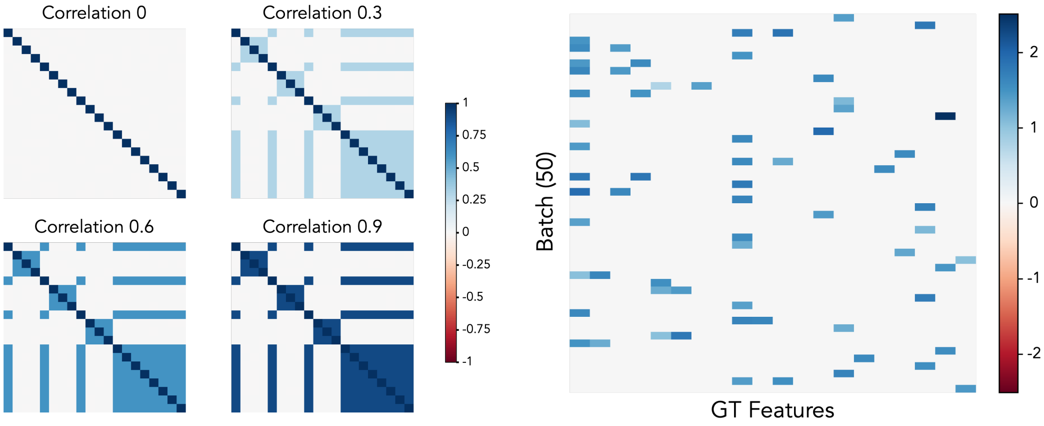
B.1.3 Detailed Results on Synthetic Data

In this section, we provide detailed results from the synthetic experiments (Figure 4). Figure 11a displays the ground truth correlation matrices used to construct dictionaries with varying levels of intra-level correlation; we focus here on four representative settings: 0, 0.3, 0.6, and 0.9.

All models are evaluated using a fixed sparse code $\boldsymbol{z}$, shown in Figure 11b, which depicts the ground truth activation matrix $\boldsymbol{z} \in \mathbb{R}^{N \times p}$. When varying the degree of intra-level correlation, Vanilla SAE and BatchTopK consistently exhibit feature absorption. Matryoshka SAE sometimes alleviates this issue but does so by suppressing intra-level correlations entirely. In contrast, MP-SAE reliably recovers both the hierarchical structure and intra-level correlations at low to moderate levels (0, 0.3) and at high correlation levels (0.9), it prioritizes preserving the hierarchy over matching correlation. We summarize our detailed observations for each correlation level below:

- **Figure 12 (Correlation = 0):** We observe feature absorption in both the Vanilla SAE and BatchTopK models. Specifically, features corresponding to child nodes are often aligned with those of their parent nodes, as evidenced by prominent horizontal structures in the second row of Figure 12. This indicates that the recovered child features (2–4,6–8,10–11) are not disentangled from their respective parents (1,5,9). While the Matryoshka SAE occasionally mitigates this issue (see rightmost runs in the Matryoshka column of Figure 12), it does so by introducing negative correlations between features at the same level, which can result in partial misalignment. In contrast, MP-SAE perfectly recovers the ground truth dictionary without exhibiting feature absorption. In terms of support recovery and sparse code estimation, all methods (excluding the failed runs of Matryoshka in rows 2 and 4) correctly identify the active set. However, in Vanilla SAE and BatchTopK, feature absorption leads to underestimation of parent activation values when a child is active. When successful, Matryoshka mitigates this issue. MP-SAE achieves exact recovery, with the inferred sparse codes $\hat{\boldsymbol{z}}$ matching the ground truth $\boldsymbol{z}$ both in support and magnitude.

- **Figure 13 (Correlation = 0.3):** Feature absorption remains a consistent issue in both Vanilla SAE and BatchTopK. Nonetheless, these models partially reflect the ground truth intra-level correlations among the green parent nodes without children (13–20). Matryoshka SAE continues to exhibit inconsistent behavior: while it occasionally reduces feature absorption, it fails to consistently capture intra-level dependencies. In contrast, MP-SAE maintains perfect recovery, accurately reconstructing the sparse codes with $\hat{\boldsymbol{z}} = \boldsymbol{z}$ and showing no evidence of feature absorption.
- **Figure 14 (Correlation = 0.6):** Vanilla SAE and BatchTopK partially capture intra-level correlations, with correlation values closest to the ground truth among the blue child nodes. However, these correlations remain weaker or less accurate for other node types. Matryoshka SAE continues to struggle, failing to recover meaningful intra-level dependencies. In contrast, MP-SAE achieves partial recovery of intra-level correlations—particularly among parent nodes without children—and more faithfully preserves the hierarchical organization of the dictionary through its inductive bias toward conditional orthogonality.
- **Figure 15 Correlation = 0.9):** At high levels of intra-level correlation, all SAEs exhibit degraded performance in recovering features. Vanilla SAE and BatchTopK still capture some similarity among the last-layer children but tend to flatten the hierarchy by focusing on correlated low-level features while neglecting higher-level ones. MP-SAE, interestingly, separates each hierarchical level into two, resulting in four orthogonal sublevels composed of 1, 10, 3, and 6 features, respectively. Within each ground-truth level of correlated features, MP-SAE learns one anchor direction capturing the shared structure, and additional orthogonal directions that describe deviations from this anchor to reconstruct the remaining features, as observed in the correlation matrices $\boldsymbol{D}^\top \boldsymbol{D}$. Since it is not trained with a fixed target sparsity, this separation is achieved by increasing the support: most inputs now activate 2–4 components compared to the ground-truth sparsity of 1–2. This behavior preserves conditional structure even under strong entanglement—at the cost of reduced correlation—illustrating MP-SAE's bias toward maintaining hierarchical organization when representations are highly coupled.



(a) Ground truth correlation matrices used in the experiments, corresponding to four levels of intra-level correlation: 0, 0.3, 0.6, and 0.9.

(b) Ground truth reference activation matrix $\boldsymbol{z}$ used for generating synthetic inputs in the following experiments.

Figure 11: **Ground truth structures used for the synthetic experiments.** (a) shows the correlation levels introduced among dictionary elements at each setting. (b) shows the corresponding ground truth activation patterns used to construct the inputs.

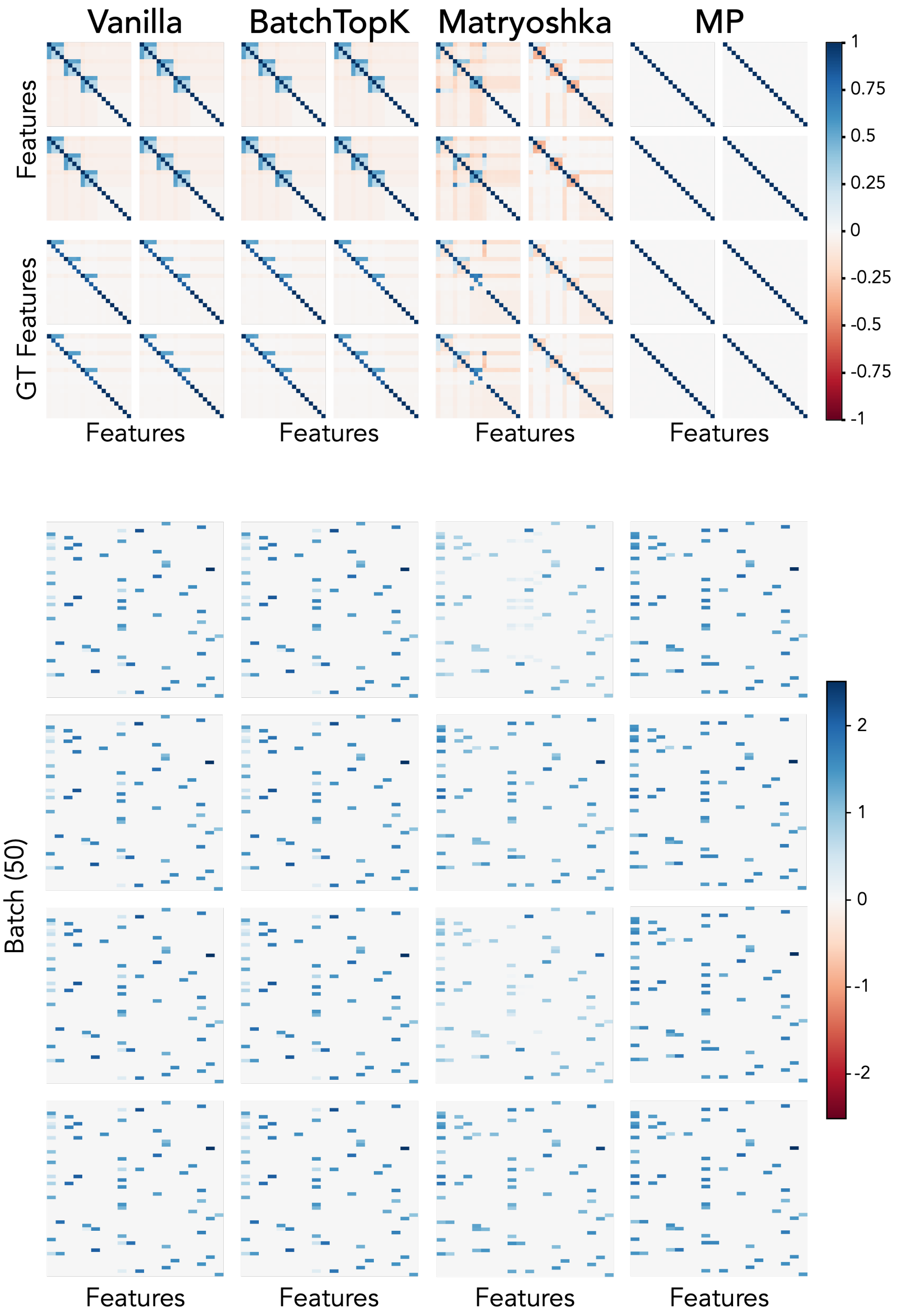


Figure 12: **4 runs for correlation level 0. Top:** Correlation matrices — the top row shows $\boldsymbol{D}^\top \boldsymbol{D}$ (self-similarity of learned features), and the bottom row shows $\boldsymbol{D}_{\mathrm{GT}}^\top \boldsymbol{D}$ (alignment with ground truth). **Bottom:** Recovered sparse codes $\mathbf{z}$.

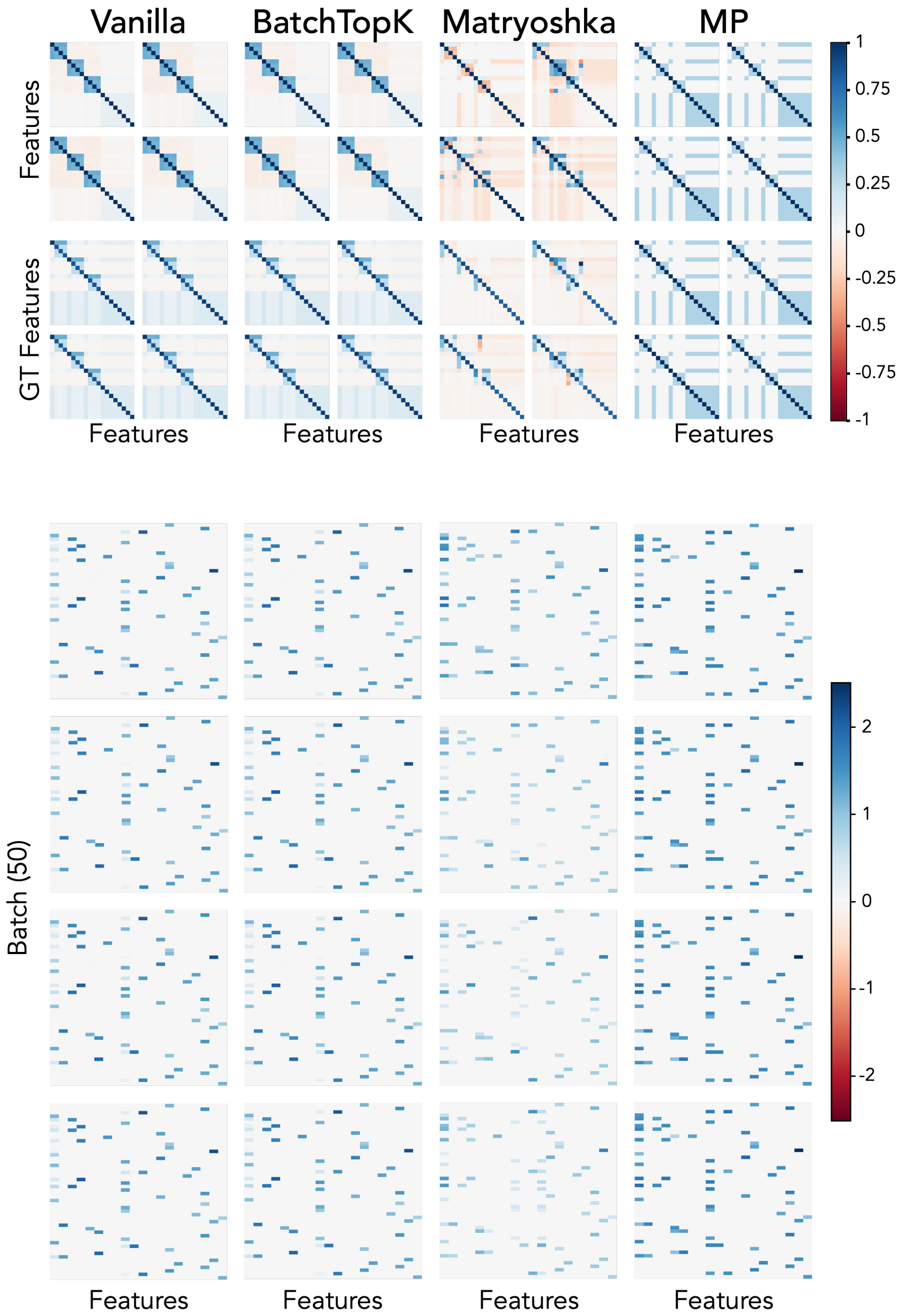


Figure 13: **4 runs for correlation level 0.3. Top:** Correlation matrices — the top row shows $\boldsymbol{D}^\top \boldsymbol{D}$ (self-similarity of learned features), and the bottom row shows $\boldsymbol{D}_{\text{GT}}^\top \boldsymbol{D}$ (alignment with ground truth). **Bottom:** Recovered sparse codes $\mathbf{z}$.

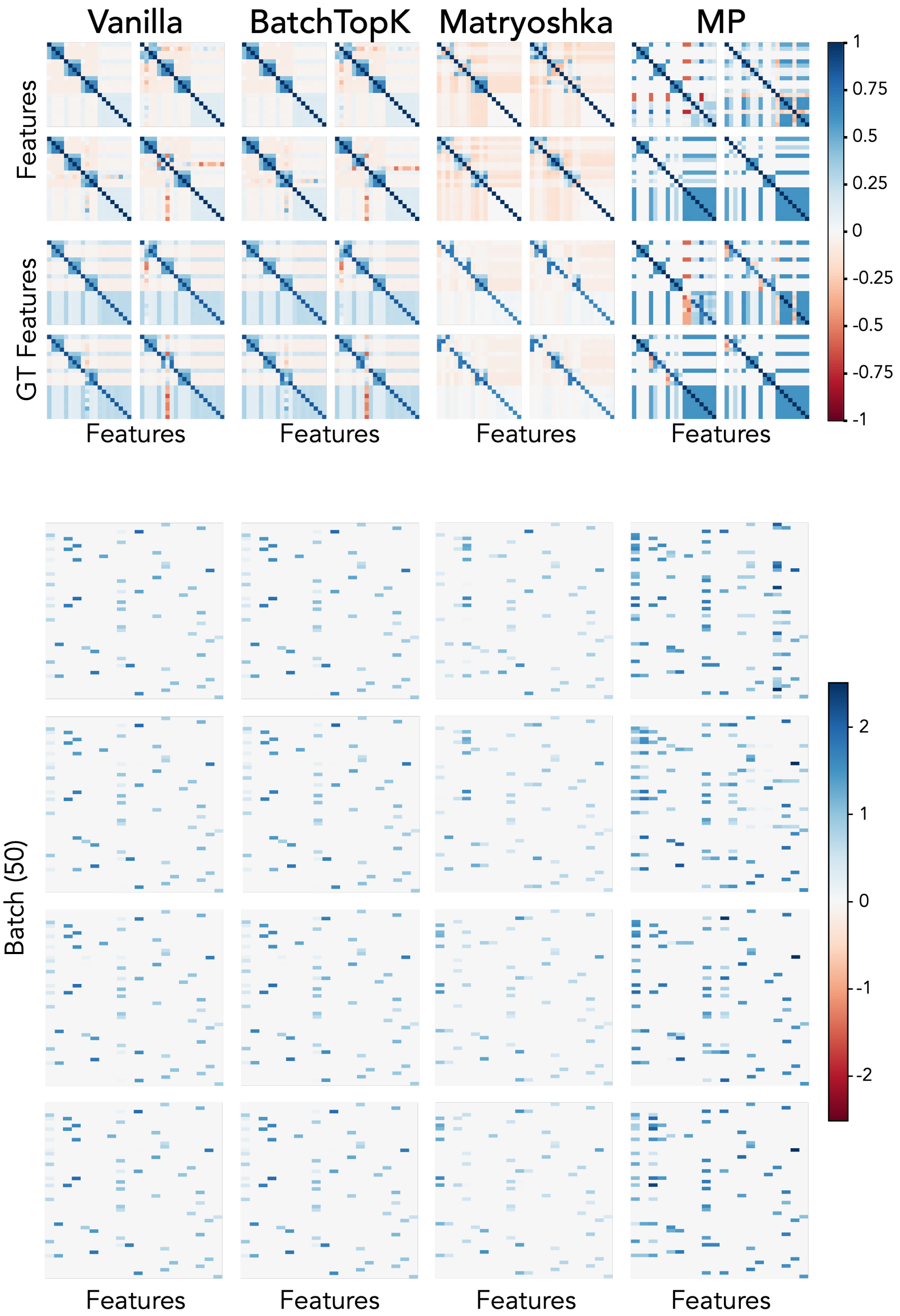


Figure 14: **4 runs for correlation level 0.6. Top:** Correlation matrices — the top row shows $\boldsymbol{D}^\top \boldsymbol{D}$ (self-similarity of learned features), and the bottom row shows $\boldsymbol{D}_{\mathrm{GT}}^\top \boldsymbol{D}$ (alignment with ground truth). **Bottom:** Recovered sparse codes $\mathbf{z}$.

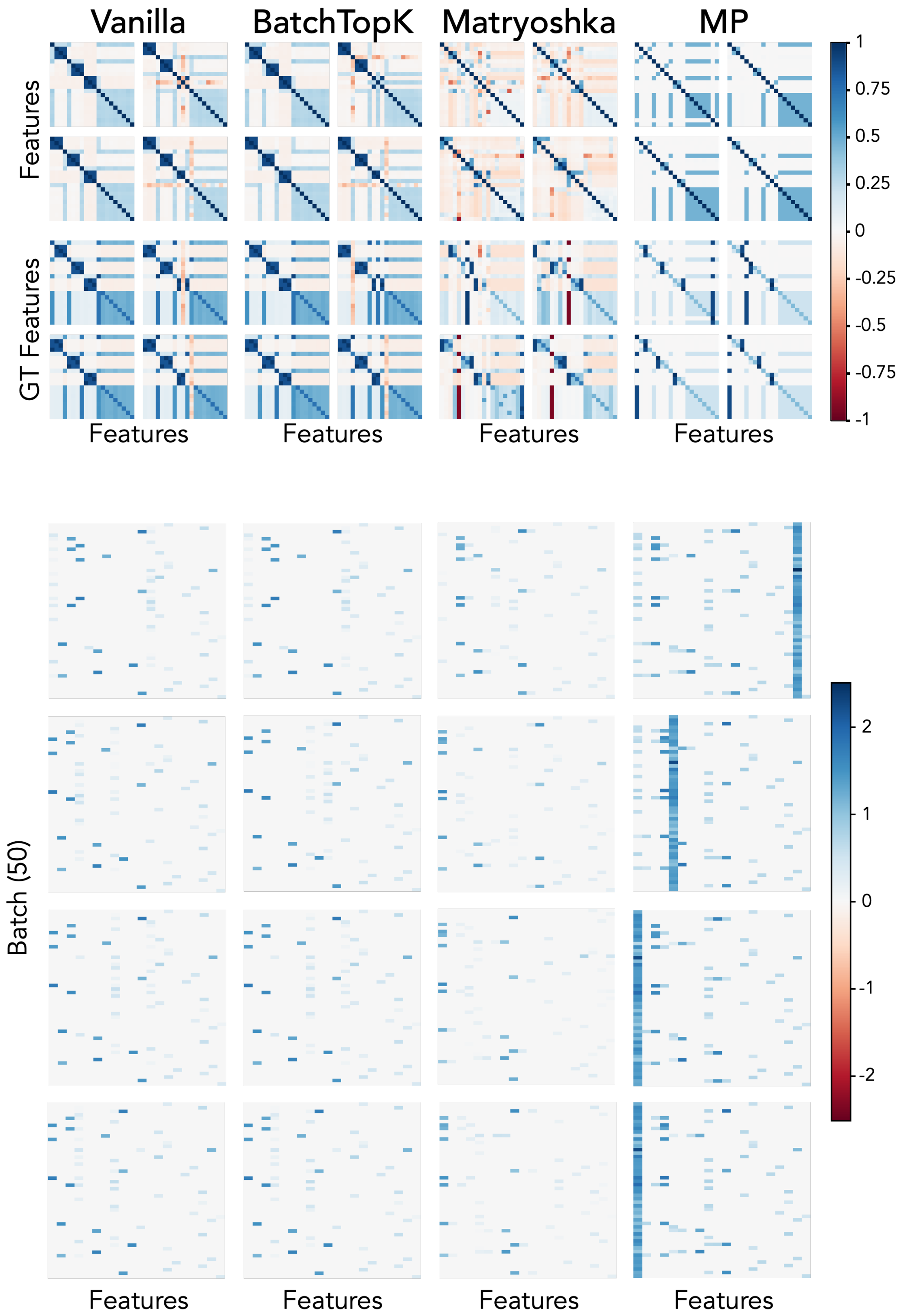


Figure 15: **4 runs for correlation level 0.9. Top:** Correlation matrices — the top row shows $\boldsymbol{D}^\top \boldsymbol{D}$ (self-similarity of learned features), and the bottom row shows $\boldsymbol{D}_{\mathrm{GT}}^\top \boldsymbol{D}$ (alignment with ground truth). **Bottom:** Recovered sparse codes $\mathbf{z}$.

B.1.4 Introducing Variance in Firings

In the original codebase of Bussmann et al. [20], activation magnitudes are fixed across all inputs. This implicitly assumes that parent and child firings are perfectly correlated:

$$z_p = \lambda z_c.$$

Such a configuration defines a strict hierarchical setting that inherently favors *feature absorption*. In contrast, our formulation does not impose any constraint on the correlation between firing magnitudes across concepts. Our setup generalizes the original formulation by removing the perfect correlation assumption. This is achieved by introducing variance in the activation magnitudes while maintaining the sole structural assumption that a child feature can only be active if its parent is active (Definition 4.1).

Rationale. To illustrate the consequences of assuming perfect correlation in activation magnitudes, consider two concepts: *"red object"* (parent) and *"apple"* (child). The child can only activate when the parent does (assuming all apples are red). If the parent's firing magnitude z_p represents color intensity and the child's z_c represents size, assuming z_p and z_c are perfectly correlated is equivalent to stating that larger apples are always redder. This perfectly merges the two informational components—color intensity ($z_p \boldsymbol{D}_p$) and apple size ($z_c \boldsymbol{D}_c$)—into a single absorbed direction $z_p \boldsymbol{D}_p + z_c \boldsymbol{D}_c$, thereby eliminating their hierarchical distinction. In this extreme case, it becomes unclear whether the absorbed feature should still be interpreted as a child, since its activation is perfectly predictable from the parent. We argue that imperfect correlation between z_p and z_c is a necessary condition for preserving hierarchical interpretability.

Implications for Sparse Methods. Following Occam's razor, an ℓ_0-penalized method such as MP-SAE will optimally encode both parent and child with a single feature when their activations are nearly identical, since the absorbed feature forms a highly compressible direction [143]. Encoding the absorbed feature $z_p \boldsymbol{D}_p + z_c \boldsymbol{D}_c$ yields a sparsity of 1, whereas disentangling parent and child requires a sparsity of 2. Therefore, it is optimal for MP-SAE to represent the parent and child through a single absorbed feature when it can perfectly reconstruct $\boldsymbol{x}$ with lower sparsity, which is precisely what occurs when no firing variance is introduced, as in the original benchmark. However, because of the greedy nature of Matching Pursuit, MP-SAE may still learn to disentangle the parent and child, depending on initialization. Figure 16 illustrates this phenomenon: it shows the correlation matrices and ground-truth alignment for the unmodified Matryoshka SAEs benchmark, trained for 15,000 iterations with a threshold of $T = 0.6$ for MP-SAE.

Additional results with different child firing distributions. In our generative data process, both parent and child activations initially follow the same distribution, $z_p \sim \mathcal{N}(1.5, {}^1\!/\!{}_{4^2})$ and $z_c \sim \mathcal{N}(1.5, {}^1\!/\!{}_{4^2})$. However, high variance combined with equal means ($\mu_p = \mu_c$) can produce edge cases where a strong child and a weak parent yield $\boldsymbol{x} = z_p \boldsymbol{D}_p + z_c \boldsymbol{D}_c \approx z_c \boldsymbol{D}_c$, effectively making the parent contribution negligible and thus deviating from Definition 4.1.

To assess the robustness of MP-SAE under varying child activation statistics, we evaluated its performance across a grid of child means and variances, while keeping the parent distribution fixed as $z_p \sim \mathcal{N}(1, {}^1\!/\!{}_{4^2})$. For each configuration, we computed an *absorption score*, defined as the average cosine similarity between the learned child features and their corresponding ground-truth parents over 15 independent runs. Lower similarity indicates successful disentanglement, whereas higher similarity reflects feature absorption. NaN values correspond to cases where the model failed to recover the child feature due to its very low firing amplitude. All activations were constrained to remain positive when active (entries marked "–" in Table 1 denote untested configurations).

Across all evaluated conditions, MP-SAE consistently achieves low absorption scores, confirming its robustness and its ability to recover hierarchical structure even under challenging firing regimes. Indeed, when the child average firing magnitude is particularly low (0.1), MP-SAE remains the only method capable of recovering the hierarchical structure, owing to its iterative greedy encoder. Interestingly, when the variance of the child distribution decreases for a mean of 0.5—a regime where overall performance becomes comparable to Matryoshka—the two methods exhibit opposite trends: Matryoshka performs better at low variance, whereas MP-SAE improves as variance increases. This observation suggests that the two approaches embody complementary inductive biases: Matryoshka is particularly effective in scenarios dominated by strong feature absorption and near-perfect correlation, while MP-SAE excels when moderate variance is present in the firing magnitudes.

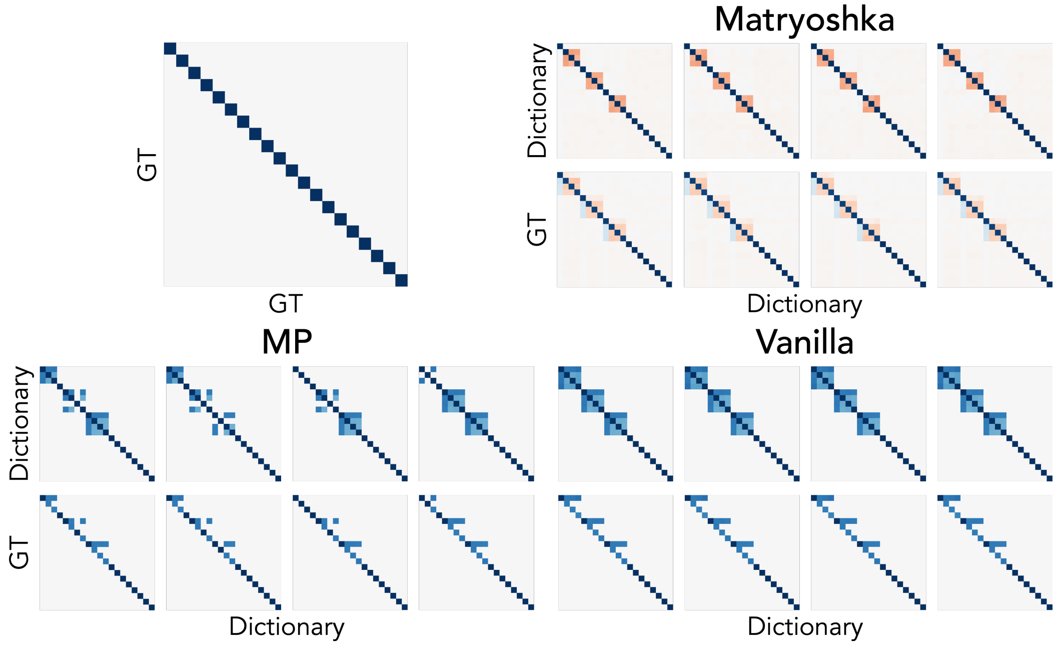


Figure 16: **Evaluating MP-SAE on the original Matryoshka benchmark[5], where firings are perfectly correlated.** For each SAE, four independent runs are shown. The top row displays $\boldsymbol{D}^\top \boldsymbol{D}$ (self-similarity of learned features), and the bottom row shows $\boldsymbol{D}_{\text{GT}}^\top \boldsymbol{D}$ (alignment with ground truth). **Top left:** ground-truth correlation matrix, where features are perfectly orthogonal. **Matryoshka** overcomes feature absorption at the cost of negative interference. **Vanilla** consistently learns absorbed features. **MP-SAE** occasionally recovers parent–child pairs thanks to its inductive bias, but under perfect correlation, its behavior is unstable.

Table 1: **Comparison of absorption scores across sparse autoencoders for different child activation distributions.** Child activations follow normal distributions with varying means (rows) and standard deviations (columns), while the parent distribution is fixed as $z_p \sim \mathcal{N}(1, {}^1/_{4^2})$. Lower absorption scores indicate better disentanglement between parent and child features. MP-SAE maintains low absorption across settings and remains the only method robust to low mean child firings.

MP	0.1	0.5	1.0
1/400	3.6e-03	7.3e-02	1.3e-03
1/40	5.0e-03	6.2e-02	1.2e-03
1/8	–	4.3e-02	1.1e-03
1/4	–	–	1.1e-03

Matryoshka	0.1	0.5	1.0
1/400	nan	4.8e-02	2.1e-01
1/40	nan	9.0e-02	1.5e-01
1/8	–	9.6e-02	2.1e-01
1/4	–	–	2.2e-01

Vanilla	0.1	0.5	1.0
1/400	nan	7.0e-01	4.5e-01
1/40	nan	7.0e-01	4.5e-01
1/8	–	7.0e-01	4.7e-01
1/4	–	–	4.7e-01

BatchTopK	0.1	0.5	1.0
1/400	nan	5.0e-01	4.8e-01
1/40	nan	5.4e-01	4.5e-01
1/8	–	4.9e-01	4.6e-01
1/4	–	–	4.2e-01

[5] https://github.com/noanabeshima/matryoshka-saes

B.2 Large Vision Experiment

We train our vision SAEs on activations from four pretrained vision and VLM models: CLIP, DINOv2, SigLIP, and ViT. Representations are extracted from the final layer of each model, utilizing all spatial tokens (e.g., DINOv2 yields approximately 261 tokens per image). Each SAE processes these token-wise activations independently. Training is conducted on the ImageNet-1K training set, comprising around 1,3 Millions images. Over 50 epochs, this results in approximately $1.3 \times 10^6 \times 50 \times 261 \approx 16.9$ billion input tokens (for DINOv2). We employ a batch size of 8,000 tokens per step and train all models using the AdamW optimizer with a cosine learning rate schedule: the learning rate warms up from 10^{-6} to 5×10^{-4} and decays back to 10^{-6} by the final epoch. A fixed weight decay of 10^{-5} is applied throughout. All SAEs utilize an expansion factor of 25, meaning the learned dictionary $\boldsymbol{D} \in \mathbb{R}^{c \times d}$ satisfies $c = 25d$, where d is the dimensionality of the input activations. Each column $\boldsymbol{D}_i$ is constrained to lie on the unit ℓ_2 ball: $\|\boldsymbol{D}_i\|_2 \leq 1$. The loss given is the standard MSE. To maintain active support coverage, a revive factor of 10^{-5} is added to any pre-code unit that fails to activate in a given batch, slightly increasing its pre-activation to reintroduce gradient flow. For Vanilla SAEs, we apply an adaptive ℓ_1 penalty: if the empirical ℓ_0 sparsity of a batch exceeds a target threshold, the ℓ_1 regularization weight is increased to suppress overactivation. All encoder architectures consist of a one-layer linear projection followed by a ReLU activation. For the pareto results, one SAE is trained for each configuration (sparsity, models).

B.3 Code Structure and Analysis of Dictionary

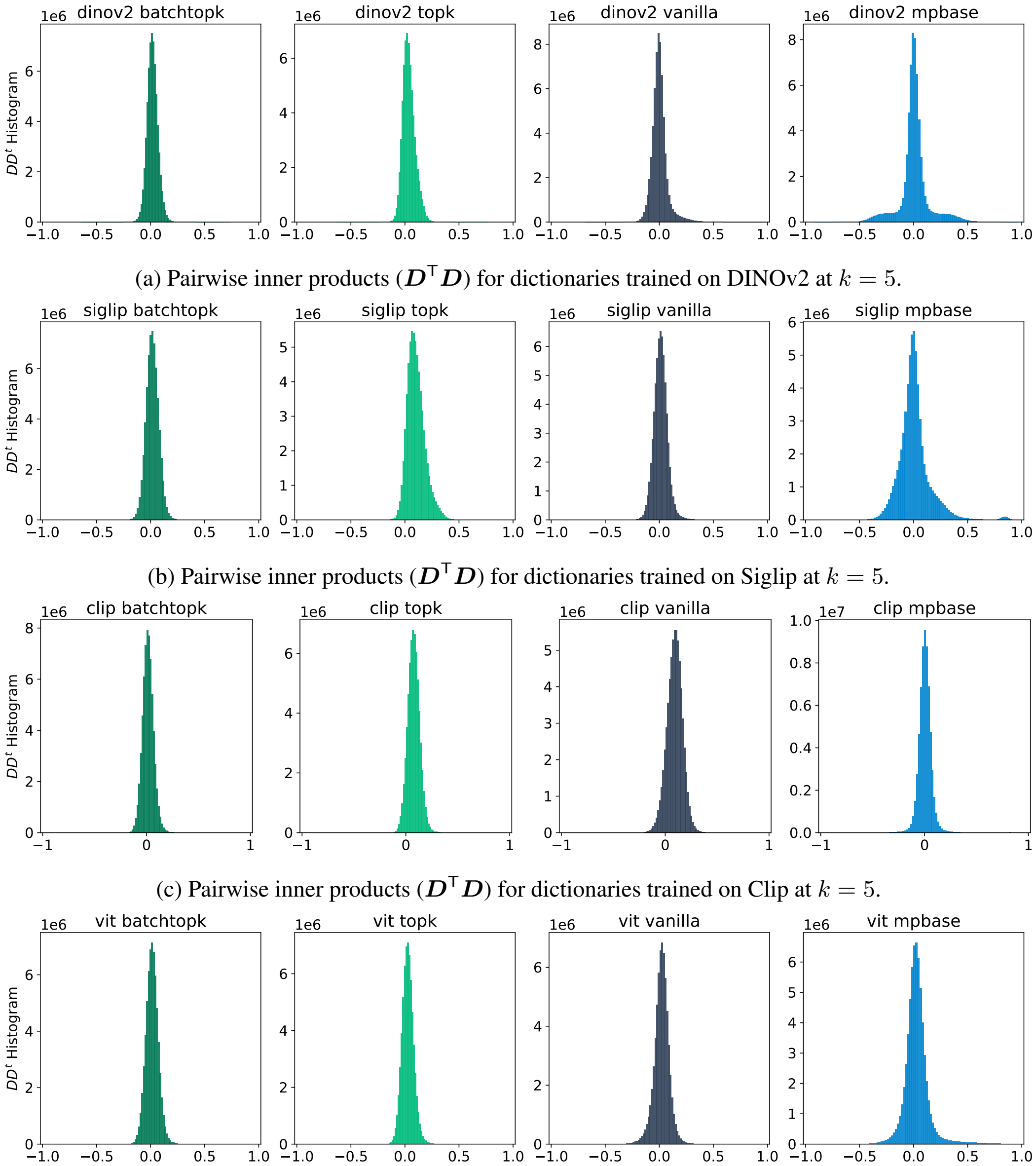
Effective Rank of Feature Co-Activation. To quantify the diversity of feature usage in sparse autoencoders, we compute the effective rank of the co-activation matrix $\boldsymbol{Z}^\top \boldsymbol{Z}$, where $\boldsymbol{Z} \in \mathbb{R}^{n \times p}$ contains the sparse codes across n inputs. Each entry in $\boldsymbol{Z}^\top \boldsymbol{Z}$ reflects how often pairs of features are jointly active, and its spectral structure reveals how concentrated or distributed these co-activations are. The effective rank [144] is defined as:

$$\mathrm{Erank}(\boldsymbol{Z}^\top \boldsymbol{Z}) = \exp\left(- \sum_{i=1}^{p} \tilde{\lambda}_i \log \tilde{\lambda}_i \right) \tag{2}$$

where $\tilde{\lambda}_i$ are the eigenvalues of $\boldsymbol{Z}^\top \boldsymbol{Z}$ normalized to sum to 1. This corresponds to the exponential of the Shannon entropy of the spectrum. High effective rank indicates that feature co-activations are spread across many directions (i.e., more diverse, less redundant usage), while low effective rank suggests repeated use of a few dominant feature combinations.

Pairwise Coherence of Dictionary Features. To further analyze the internal structure of learned dictionaries, we examine the distribution of pairwise inner products $\boldsymbol{D}^\intercal \boldsymbol{D}$, which captures the angular alignment between features. Figure 17 shows histograms ofof the values for dictionaries trained on the vision models. We used the SAEs from the pareto front, with a fixed inference-time sparsity of $k = 5$. We observe that dictionaries learned by standard SAEs (e.g., Vanilla, TopK) exhibit a distribution sharply peaked at zero, reflecting a strong bias toward global quasi-orthogonality. This is a direct consequence of their encoding mechanism: features are selected via a single global projection, which incentivizes mutually uncorrelated features to avoid interference.

By contrast, dictionaries learned by MP-SAE display broader distributions, including significant mass at nonzero values. This reflects an important difference in inductive bias: MP-SAE does not enforce orthogonality globally. Instead, its sequential, residual-guided encoder dynamically selects features that are orthogonal *conditioned* on prior selections. This means that the dictionary can afford to contain closely aligned features – as long as they do not co-activate for the same input.



(a) Pairwise inner products ($\boldsymbol{D}^\top \boldsymbol{D}$) for dictionaries trained on DINOv2 at $k = 5$.

(b) Pairwise inner products ($\boldsymbol{D}^\top \boldsymbol{D}$) for dictionaries trained on Siglip at $k = 5$.

(c) Pairwise inner products ($\boldsymbol{D}^\top \boldsymbol{D}$) for dictionaries trained on Clip at $k = 5$.

(d) Pairwise inner products ($\boldsymbol{D}^\top \boldsymbol{D}$) for dictionaries trained on ViT at $k = 5$.

Figure 17: Distribution of dictionary coherence ($\boldsymbol{D}^\top \boldsymbol{D}$) reveals differing inductive biases. Standard SAEs encourage globally orthogonal features; MP-SAE tolerates correlated features and resolves interference at inference.

B.4 Multimodal Models

Setup. We evaluates SAEs of 4 VLMs: CLIP, SigLIP, SigLIP2, and AIMv2. We use the final layer of the shared embedding space—that is, the common representation into which both image and caption inputs are projected. This layer reflects the aligned modality-invariant space optimized by contrastive pretraining. We train all SAEs on the full MS-COCO dataset [102], which consists of approximately 100,000 images and 500,000 associated captions. Each training example corresponds to a single embedding vector (either image or text), and models are trained jointly across both modalities using a shared dictionary (expansion factor 25). As in prior settings, we constrain each dictionary column to lie on the unit ℓ_2 ball. All SAEs use a one-layer encoder followed by ReLU activations and are trained with mean squared error (MSE) loss. We fix the target inference-time sparsity to $\|\boldsymbol{z}\|_0 = 5$. The optimization procedure mirrors that used for vision models for a total of 20 training epochs. The same revive mechanism is applied, whereby inactive units are nudged via a small additive bias (10^{-5}) to encourage gradient flow.

Modality score To quantify the extent to which learned features specialize by modality or capture shared structure, we compute the Modality Score for each feature, following the formulation of [105]. Values near 1 indicate image-specific features; values near 0 indicate text-specific features; and intermediate values reflect balanced, multimodal activation. To ensure that Modality Scores reflect relative activation patterns rather than absolute energy differences between modalities, we scale the energy of text inputs by a factor of $1/5$ before computing the above quantities. This corrects as we have 5 times more captions than images. We find that the same overall patterns emerge when we apply a modality wise normalization.

The figure 18 illustrates the modality gap and how different inference strategies shape learned representations in vision-language models. Each point on the sphere represents a direction in a shared embedding space, such as the one learned by CLIP [75], where both images and captions are aligned.

On the left, standard sparse autoencoders (SAEs) tend to learn split dictionaries [103, 104], assigning different features to each modality despite their alignment in the joint space. This results in high modality selectivity: visual and textual inputs activate disjoint sets of features.

On the right, MP-SAE progressively refines its representation by explaining modality-specific components in early steps and shifting focus to shared structure in later steps. This allows the model to discover genuinely multimodal concepts that respond to both image and text inputs, effectively bridging the modality gap and producing more balanced, semantically aligned features.

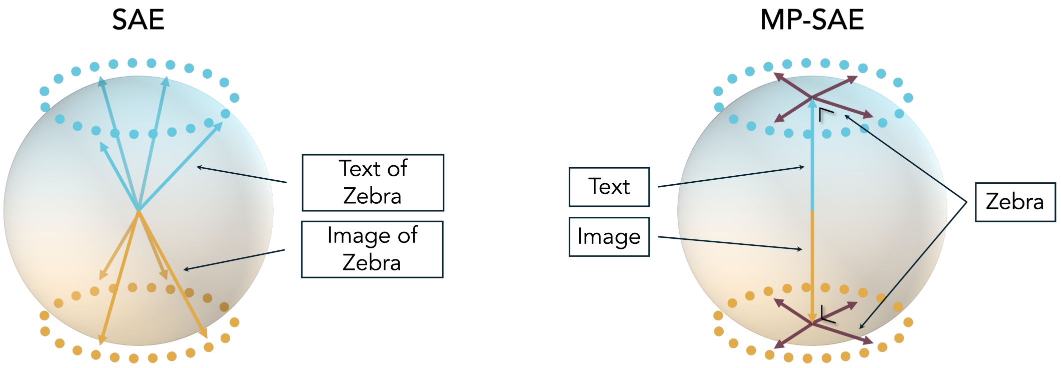


Figure 18: **Illustration of modality-selective vs. multimodal inference.** Each point represents an image or caption embedding on the unit sphere of a joint representation space. Left: standard SAEs learn split dictionaries, with features specializing in only one modality (blue = text, yellow = image), despite semantic alignment. Right: MP-SAE explains modality-specific content in early steps and aligns shared concepts in later ones. For instance, an image of a zebra and the text "a zebra in the savannah" may initially activate modal specific features, but converge toward shared features encoding the concept "zebra."

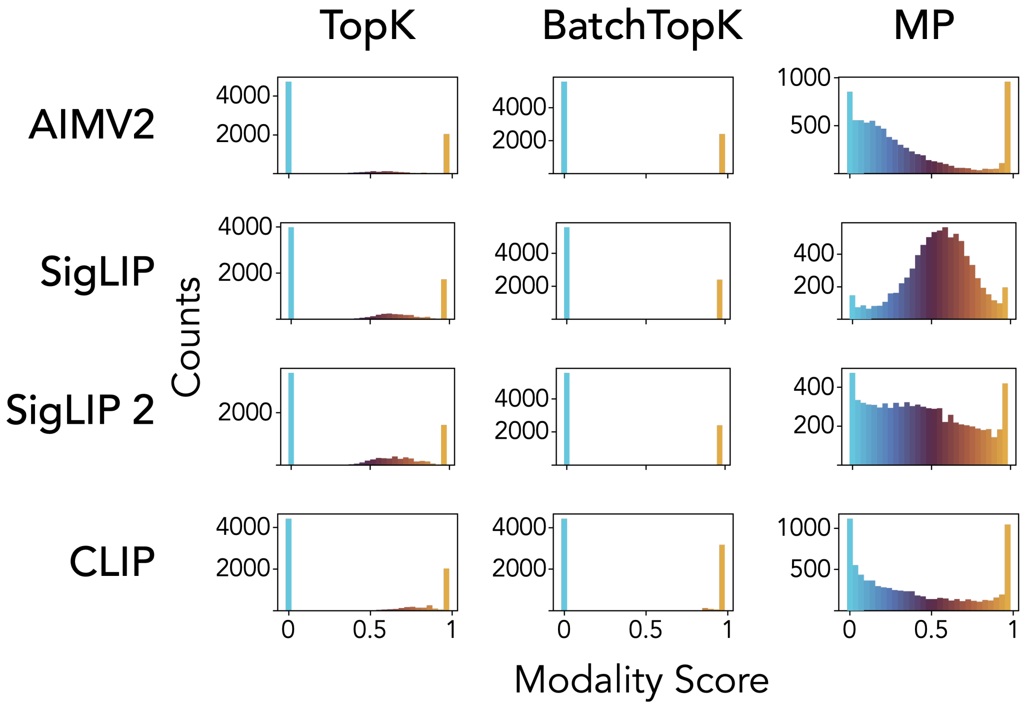


Figure 19: Distribution of modality scores for features learned by different SAEs, across four vision-language models: CLIP, SigLIP, SigLIP2, and AIMv2. Standard SAEs (e.g., TopK, BatchTopK) tend to produce bimodal distributions, with many units activating exclusively for image or text inputs, indicating a strong modality-specific specialization. In contrast, MP-SAE yields a significantly flatter distribution with substantial density near 0.5, revealing the emergence of genuinely multimodal features that respond to both modalities. This supports the hypothesis that MP-SAE's inference allows it to progressively isolate shared semantic structure after accounting for modality-specific variance.

B.5 Preliminary Experiments with Language Models

B.5.1 Experimental details

To assess whether our analysis of the vision domain (see Sec. B.4) generalizes to a language modeling setting, we perform preliminary experiments using a 4-layer Transformer model [145] pretrained on the TinyStories dataset [146].[6] We train the Vanilla [14], TopK [15], and MP-SAE (see Sec. 3) on residual stream representations extracted from midway through the model, i.e., end of the second Transformer block (from amongst 4 total blocks). p, which refers to the width of our SAEs, i.e., the size of the sparse code z, is set to be $4 \times m$ for all SAEs; here, m corresponds to the dimensionality of the residual stream. For TopK and MP-SAE, we set $\ell_0 = 100$ for any inputted representation; for Vanilla SAE, we search for a value of λ to ensure after training ℓ_0, on average, is 100. Inline with prior work [22], activations are standardized by computing a population mean and standard deviation and rescaled to be, on expectation, $\sqrt{m}$ magnitude in norm. Training is performed using Adam optimizer [147], with a constant learning rate of 10^{-3}, $\beta_1, \beta_2 = 0.9, 0.95$, weight decay of 10^{-4}, and gradient clipping at unit-norm magnitude for all the SAEs. Batch-size is set to be 5000 tokens, each of which is randomly sampled from samples (stories) from the train-split of TinyStories. Training goes on for approximately 40K iterations. All results in the following experiments are performed on samples drawn from the eval-split of TinyStories.

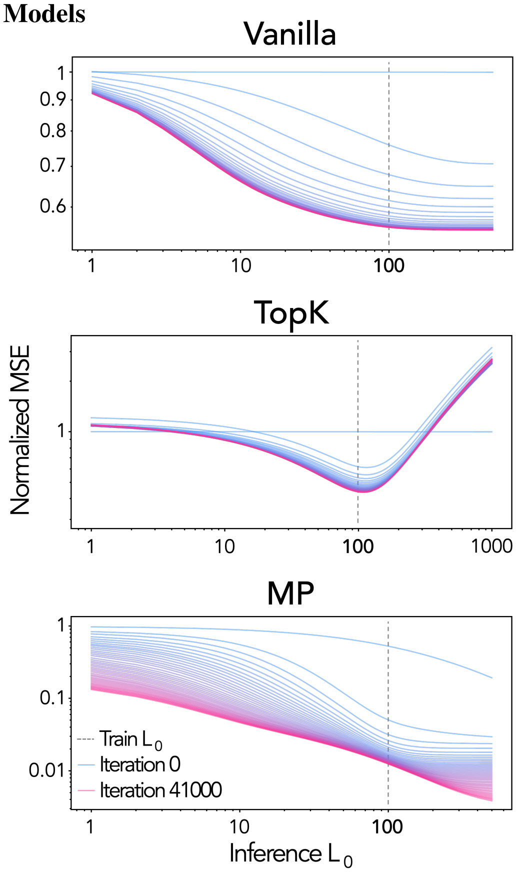


Figure 20: **Normalized MSE with increase in training iterations and inference-time sparsity.** We train Vanilla, TopK, and MP SAEs and plot the normalized MSE between their predicted vs. ground-truth representations from models trained on TinyStories. Results clearly show the flexibility of MP-SAE: increasing number of latents used for reconstruction monotonically reduce error, allowing one to use more or less latents, as desired. This property is emergent with training, however, since earlier iterations of MP-SAE do not exhibit it.

B.5.2 Results

Reconstruction Error with Inference L$_0$. Similar to Fig. 8, for all SAEs, we take their parameters trained up to iteration t and compare the inputted representations (x) with the ones predicted by the SAE given its k largest magnitude latents in the sparse code, denoted $x_{t,k}$. Specifically, we compute the **normalized MSE**, $\mathbb{E}_x\left[{||\hat{x}_{t,k}-x||^2}/{||x||^2}\right]$, for increasing values of t and k. We note this experiment is different from the one reported in Fig. 8 because it also assesses the effects of training, hence going beyond highlighting the inference-time flexibility of MP-SAE. Such an analysis is manageable in the current setting since models are not exceptionally big ($\sim$ 1M parameters).

Results are shown in Fig. 20, where we most noticeably see a similar trend as Fig. 8: as k is increased, Vanilla SAEs smoothly move towards a minimum amount of error that corresponds to the value of k used for training; TopK SAEs see reduction in error up to the value of k used for training, witnessing a large increase in error thereafter; and MP-SAEs again exhibit their flexibility, with error monotonically decreasing even beyond the value of k used for training. It is worth noting that this flexibility of MP-SAE only emerges after enough training has occurred—in fact, earlier checkpoints exhibit a saturation or lower-bound of loss akin to the Vanilla SAEs. Finally, we emphasize again that this inference-time flexibility is a natural consequence of building on error residuals, unlike prior SAEs. Prior work hoping to elicit such a behavior had to manually train via a mixture of sparsity values [15].

[6] We also provide a trained MP-SAE on Gemma activations at `https://github.com/eslubana/mpsae`.

Babel Score Analysis on Language Models. Following the analysis in Figure 7, we compute the Babel scores for dictionaries learned on language model embeddings using MP, Vanilla, and TopK SAEs. Results are shown in Figure 21, reporting both the coherence of the full dictionary (left) and that of the concepts co-activated at inference (right).

The same trends observed in the vision models hold here as well. MP-SAE learns dictionaries with higher Babel scores globally—indicating more interference across features—but selects sets of concepts with lower mutual interference at inference time. This again reflects MP's bias toward conditional orthogonality. Conversely, Vanilla and TopK SAEs learn more globally incoherent dictionaries, but their inference-time selections often include more correlated features. Notably, the y-axis ranges in Figure 21 are similar to those from the vision experiments 7, suggesting that this behavior is not modality-specific indicating that this effect is not data dependent.

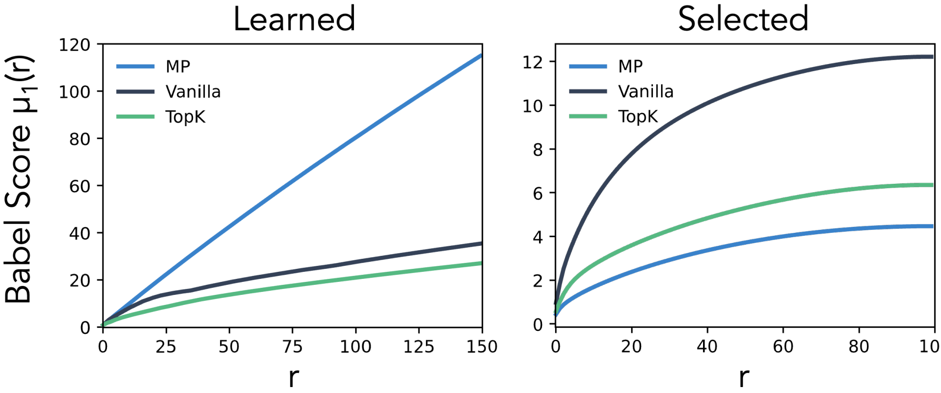


Figure 21: **Babel score analysis on language models.** We report the coherence of the full learned dictionary (left) and the subset of concepts co-activated at inference (right), across MP, Vanilla, and TopK SAEs. As in the vision setting, MP learns globally coherent dictionaries but selects less interfering concepts at inference, while Vanilla and TopK show the opposite pattern. The similarity in value ranges across modalities suggests that these effects are not data dependent.

Effective Rank. We compute the effective rank of the co-activation matrix $Z^\top Z$ for MP, ReLU, and TopK SAEs trained on language model embeddings.

As shown in Figure 22, MP-SAE exhibits a non-monotonic trend: the effective rank increases with sparsity, peaks at a critical point, and then declines. This suggests that MP initially promotes diverse feature use before redundancy emerges. In contrast, ReLU shows a steady decline and eventually saturates, while TopK decays slowly but drops sharply when the inference ℓ_0 exceeds the training level. The drop in effective rank reflects increasing feature reuse and reduced diversity in the representation.

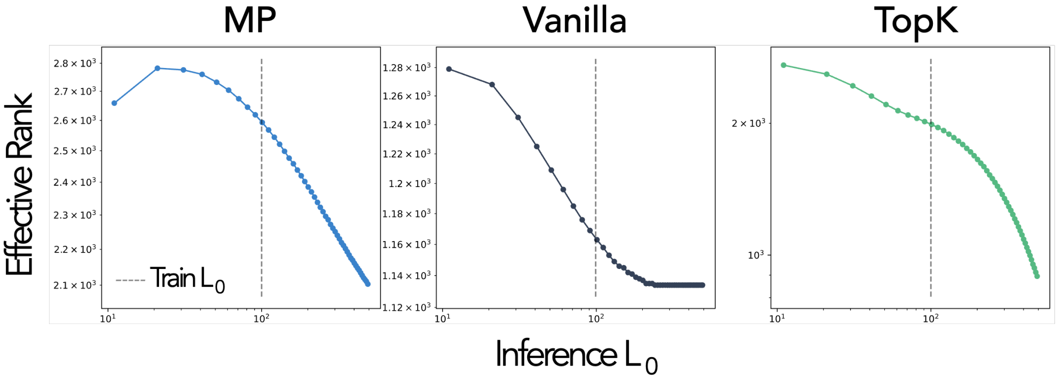


Figure 22: **Effective rank of the co-activation matrix $Z^\top Z$.** MP-SAE shows a rise-then-drop trend, indicating initial diversification followed by redundancy. ReLU steadily declines and saturates, while TopK decays slowly but drops sharply when inference sparsity exceeds training.

C Theoretical Guarantees for MP-SAE Inference

We restate three foundational properties of Matching Pursuit—originally established in the sparse coding literature [74]—and interpret them in the context of sparse autoencoders. These properties help elucidate the structure and dynamics of the representations learned by MP-SAE.

- **Stepwise orthogonality** (Proposition C.1): at each iteration, the residual becomes orthogonal to the feature most recently selected by the greedy inference rule. This sequential orthogonalization mechanism gives rise to a locally disentangled structure in the representation and reflects the conditional independence induced by MP-SAE inference.
- **Monotonic decrease of residual energy** (Proposition C.2): the ℓ_2 norm of the residual decreases whenever it retains a nonzero projection onto the span of the dictionary. This guarantees that inference steps lead to progressively refined reconstructions, and enables sparsity to be adaptively tuned at inference time without retraining.
- **Asymptotic convergence** (Proposition C.3): in the limit of infinite inference steps, the reconstruction converges to the orthogonal projection of the input onto the subspace defined by the dictionary. Thus, MP-SAE asymptotically recovers all structure that is representable within its learned basis.

Proposition C.1 (Stepwise Orthogonality of MP Residuals)**.** *Let $\boldsymbol{r}^{(t)}$ denote the residual at iteration t of MP-SAE inference, and let $j^{(t-1)}$ be the index of the feature selected at step $t-1$. If the column $j^{(t-1)}$ of the dictionary $\boldsymbol{D}$ satisfy $\|\boldsymbol{D}_{j^{(t-1)}}\|_2 = 1$, then the residual becomes orthogonal to the previously selected feature:*

$$\boldsymbol{D}_{j^{(t-1)}}^{\top}\boldsymbol{r}^{(t)} = 0.$$

Proof. This follows from the residual update:

$$\boldsymbol{r}^{(t)} = \boldsymbol{r}^{(t-1)} - \boldsymbol{D}_{j^{(t-1)}}\boldsymbol{z}_{j^{(t)}}^{(t-1)},$$

with $\boldsymbol{z}_{j^{(t)}}^{(t-1)} = \boldsymbol{D}_{j^{(t-1)}}^{\top}\boldsymbol{r}^{(t-1)}$. Taking the inner product with $\boldsymbol{D}_{j^{(t-1)}}$ gives:

$$\boldsymbol{D}_{j^{(t-1)}}^{\top}\boldsymbol{r}^{(t)} = \boldsymbol{D}_{j^{(t-1)}}^{\top}\boldsymbol{r}^{(t-1)} - \|\boldsymbol{D}_{j^{(t-1)}}\|^2\boldsymbol{z}_{j^{(t)}}^{(t-1)} = \boldsymbol{z}_{j^{(t)}}^{(t-1)} - \boldsymbol{z}_{j^{(t)}}^{(t-1)} = 0. \qquad \square$$

This result captures the essential inductive step of Matching Pursuit: each update removes variance along the most recently selected direction, producing a residual that is orthogonal to it. Applied iteratively, this localized orthogonality promotes the emergence of conditionally disentangled structure in MP-SAE. In contrast, other sparse autoencoders lack this stepwise orthogonality mechanism, which helps explain the trend observed in the Babel function during inference in Figure 7.

Proposition C.2 (Monotonic Decrease of MP Residuals)**.** *Let $\boldsymbol{r}^{(t)}$ denote the residual at iteration t of MP-SAE inference, and let $\boldsymbol{z}_{j^{(t)}}^{(t)}$ be the nonzero coefficient selected at that step, Then the squared residual norm decreases monotonically:*

$$\|\boldsymbol{r}^{(t+1)}\|_2^2 - \|\boldsymbol{r}^{(t)}\|_2^2 = -\|\boldsymbol{D}_{j^{(t)}}\boldsymbol{z}_{j^{(t)}}^{(t)}\|_2^2 \leq 0.$$

Proof. From the residual update:

$$\boldsymbol{r}^{(t+1)} = \boldsymbol{r}^{(t)} - \boldsymbol{D}_{j^{(t)}}\boldsymbol{z}_{j^{(t)}}^{(t)},$$

we can rearrange to write:

$$\boldsymbol{r}^{(t)} = \boldsymbol{r}^{(t+1)} + \boldsymbol{D}_{j^{(t)}}\boldsymbol{z}_{j^{(t)}}^{(t)}.$$

Taking the squared norm of both sides:

$$\begin{aligned}\|\boldsymbol{r}^{(t)}\|_2^2 &= \|\boldsymbol{r}^{(t+1)} + \boldsymbol{D}_{j^{(t)}}\boldsymbol{z}_{j^{(t)}}^{(t)}\|_2^2 \\ &= \|\boldsymbol{r}^{(t+1)}\|_2^2 + 2\langle\boldsymbol{r}^{(t+1)}, \boldsymbol{D}_{j^{(t)}}\rangle\boldsymbol{z}_{j^{(t)}}^{(t)} + \|\boldsymbol{D}_{j^{(t)}}\boldsymbol{z}_{j^{(t)}}^{(t)}\|_2^2.\end{aligned}$$

By Proposition C.1, the cross term vanishes:

$$\langle \boldsymbol{r}^{(t+1)}, \boldsymbol{D}_{j^{(t)}} \rangle = 0,$$

yielding:

$$\|\boldsymbol{r}^{(t)}\|_2^2 = \|\boldsymbol{r}^{(t+1)}\|_2^2 + \|\boldsymbol{D}_{j^{(t)}} \boldsymbol{z}_{j^{(t)}}^{(t)}\|_2^2. \qquad \square$$

The monotonic decay of residual energy ensures that each inference step yields an improvement in reconstruction, as long as the residual lies within the span of the dictionary. Crucially, this property enables MP-SAE to support adaptive inference-time sparsity: the number of inference steps can be varied at test time—independently of the training setup—while still allowing the model to progressively refine its approximation. This explains the continuous decay observed in Figure 8, a guarantee not provided by other sparse autoencoders.

Proposition C.3 (Asymptotic Convergence of MP Residuals)**.** *Let* $\hat{\boldsymbol{x}}^{(t)} = \boldsymbol{x} - \boldsymbol{r}^{(t)}$ *denote the reconstruction at iteration* t*, and let* $\mathbf{P}_{\boldsymbol{D}}$ *be the orthogonal projector onto* $\operatorname{span}(\boldsymbol{D})$*. Then:*

$$\lim_{t \to \infty} \|\hat{\boldsymbol{x}}^{(t)} - \mathbf{P}_{\boldsymbol{D}} \boldsymbol{x}\|_2 = \lim_{t \to \infty} \|\mathbf{P}_{\boldsymbol{D}} \boldsymbol{r}^{(t)}\|_2 = 0.$$

This convergence result is formally established in the original Matching Pursuit paper by Mallat and Zhang [74, Theorem 1]. This result implies that MP-SAE progressively reconstructs the component of $\boldsymbol{x}$ that lies within the span of the dictionary, converging to its orthogonal projection in the limit of infinite inference steps. When the dictionary is complete (i.e., $\operatorname{rank}(\boldsymbol{D}) = m$), this guarantees convergence to the input signal $\boldsymbol{x}$.

D Societal Impact

Interpretability plays a critical role in the safe and trustworthy deployment of AI systems. As large-scale models are increasingly integrated into everyday technologies and used by millions of people, the risks associated with their opaque decision-making grow substantially. Without interpretability, it becomes difficult to detect biases, failures, or unintended behaviors in these powerful systems. By enabling the extraction of structured, human-interpretable features, tools like Sparse Autoencoders help researchers and practitioners understand how models encode semantics, reasoning patterns, or social attributes. This transparency is especially important in safety-critical domains such as healthcare, legal decision-making, or education, where opaque model behavior can result in harmful or unfair outcomes. Interpretability methods also support model auditing and targeted interventions, making it possible to align AI behavior with human values. However, such tools can also be misused—for example, to reverse-engineer proprietary models or infer sensitive attributes. We emphasize that our method does not amplify these existing risks.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly motivate the study by questioning whether standard sparse autoencoders (SAEs), which are designed under the Linear Representation Hypothesis (LRH), can faithfully recover features when model representations are hierarchical, nonlinear, or multidimensional. The paper then proposes MP-SAE as an architecture that operationalizes a new inductive bias—conditional orthogonality—through a sequential matching pursuit-based inference mechanism. These claims are directly substantiated by the contributions and experiments in the main body, including synthetic benchmarks, concept organization analysis, expressivity comparisons, and results on vision-language models.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a discussion of limitations in the conclusion

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper introduces definitions and theoretical motivations (e.g., conditional orthogonality, cumulative coherence via the Babel function), but it does not contain formal theorems or proofs. Our contributions are primarily empirical and architectural rather than theoretical

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide detailed descriptions of our experimental setups, datasets, and evaluation metrics in the main text. Further implementation details, including model configurations, training procedures, and hyperparameters for each experiment, will be included in the appendix to ensure full reproducibility.

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We will release all code used in our experiments in a public GitHub repository.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We describe the experimental setup, datasets, and evaluation metrics in the main text to support the interpretation of results. Detailed training configurations, including data splits, optimizer type, learning rates, number of iterations, and model-specific hyperparameters, are provided in the appendix to ensure reproducibility and clarity.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We do not report error bars or statistical significance measures. For the tree-structured experiment, we show results across multiple instantiations to demonstrate consistency. For the other experiments, variability was not a major concern, and single-run results were sufficient to support our qualitative and comparative claims.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: We do not report detailed compute resource specifications (e.g., GPU type, memory, or runtime). However, the experiments in this paper involve lightweight models and synthetic or publicly available datasets, and can be reproduced with standard hardware (e.g., a single GPU). As such, we did not find detailed compute reporting to be critical in this case, though we can include it in the final version if needed.

   Guidelines:
   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification:

   Guidelines:
   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: A discussion of potential societal impacts, including both positive and negative aspects, will be provided in the appendix.

    Guidelines:
    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.

    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**
    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?
    Answer: [NA]
    Justification:
    Guidelines:
    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.
15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**
    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?
    Answer: [NA]
    Justification:
    Guidelines:
    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.
16. **Declaration of LLM usage**
    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.
    Answer: [NA]
    Justification: LLMs were used solely for refining writing, grammar, and clarity of exposition. They were not involved in any part of the core methodology, theoretical development, experimental design, or analysis.
    Guidelines:
    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.