Stop Probing, Start Coding: Why Linear Probes and Sparse Autoencoders Fail at Compositional Generalization

Vitória Barin Pacela[*1] **Shruti Joshi**[*1] **Isabela Camacho**[2] **Simon Lacoste-Julien**[1] **David Klindt**[3]

[1]Mila - Québec AI Institute & Université de Montréal
[2]Santa Clara University
[3]Cold Spring Harbor Laboratory
[*]Equal contribution; authors listed in alphabetical order.

Abstract

The linear representation hypothesis states that neural network activations encode high-level concepts as linear mixtures. However, under superposition, when the number of concepts exceeds the activation dimension, recovering underlying latent factors from the activations requires sparse nonlinear inference, making methods such as linear probes insufficient. In this setting,c classical sparse coding methods with per-sample iterative inference leverage compressed sensing guarantees to recover latent factors. Sparse autoencoders (SAEs), on the other hand, perform nonlinear inference, but amortise it into a fixed encoder, introducing a systematic amortisation gap. We show this gap persists as the number of training samples is increased, causing SAEs to fail under out-of-distribution (OOD) compositional shifts. Our results demonstrate that the recent OOD failures of SAEs can be attributed to amortisation failures: per-sample inference at test time substantially improves OOD performance, even when using a dictionary learned by an SAE.

1 INTRODUCTION

Understanding the internal representations of Large Language Models (LLMs) is crucial for their safe and reliable deployment. The *linear representation hypothesis* (LRH) is a foundational assumption in mechanistic interpretability, stating that a model's activations are linear mixtures of underlying concepts (Jiang et al., 2024; Park et al., 2024; Smith, 2024). It has motivated crucial progress on methods such as linear probing for concept discovery and activation steering (Turner et al., 2023; Chalnev et al., 2024).

To make this precise, let $\mathbf{z} \in \mathbb{R}^{d_z}$ denote ground-truth latent variables (concepts), and let $\mathbf{y} \in \mathbb{R}^{d_y}$ denote a model's activations. The LRH asserts that the relationship between concepts and activations is linear: $\mathbf{y} \approx \mathbf{W}\mathbf{z}$ for some matrix $\mathbf{W} \in \mathbb{R}^{d_y \times d_z}$. The goal of interpretability is to recover $\mathbf{z}$ from $\mathbf{y}$—that is, to infer which concepts are active in a given representation. When $d_z > d_y$, more concepts are encoded than there are activation dimensions, a regime known as *superposition* (Elhage et al., 2022).

However, the LRH is routinely conflated with a much stronger claim. Linear *representation*—concepts being linearly encoded in activations—does not imply linear *accessibility*—that concepts are recoverable by a linear transformation. Under superposition, where $d_z > d_y$, the system $\mathbf{y} = \mathbf{W}\mathbf{z}$ is underdetermined. Each observation $\mathbf{y}$ is consistent with infinitely many $\mathbf{z}$. Without further structure, no method (linear or nonlinear) can fully recover $\mathbf{z}$. This distinction has direct consequences for downstream tasks: a linear decision boundary in latent space can become nonlinear in activation space due to the compression (Figure 1), so that linear probes trained on $\mathbf{y}$ (Alain and Bengio, 2016) can exhibit poor generalisation in-distribution and even worse out-of-distribution. This raises a natural question:

> *Under what inference procedure can the true latent factors be robustly recovered from superposed activations, even under distribution shift?*

Sparsity is an inductive bias that can help resolve this, but it requires nonlinear inference. The mathematical feasibility of this scheme is rooted in compressed sensing theory, which shows that if the concepts are sparsely active—a reasonable assumption for natural signals—it is possible to recover the underlying high-dimensional concepts from a low-dimensional representation (Donoho, 2006b). However, this recovery is not trivial; it entails two components: a dictionary of concepts, typically found via sparse dictionary learning, and a reconstruction algorithm, known as sparse inference, to identify the active concepts for a given input. Classical compressed sensing shows that k-sparse latent codes can be recovered from $d_y = O(k \log(d_z/k))$ random measurements, but recovery is necessarily nonlinear.

The amortisation gap. A potential solution is to use *amortized* sparse nonlinear inference (Gregor and LeCun, 2010), as implemented by Sparse Autoencoders (SAEs) (Ng et al., 2011; Cunningham et al., 2023). An SAE learns a fixed encoder $r : \mathbb{R}^{d_y} \to \mathbb{R}^{d_h}$ that outputs a sparse code in a single forward pass. The alternative is *per-sample inference*, which solves an optimization problem independently for each input. The *amortization gap* is the discrepancy between the two solutions (Margossian and Blei, 2023; Kim and Pavlovic, 2021; Zhang et al., 2022; Cremer et al., 2018; Schott et al., 2021; Paiton et al., 2020; O'Neill et al., 2024).

This argument adds nuance to recent studies suggesting that linear probes trained on top of LLM activations are superior to SAEs in out-of-distribution (OOD) binary classification tasks (Kantamneni et al., 2025). We argue that the recent OOD failures of SAEs are not an indictment of the superposition hypothesis, but rather a predictable consequence of replacing principled sparse inference with a brittle, amortized encoder. Instead of discarding the powerful framework of sparse coding, we embrace the geometric consequences of superposition and utilize methods equipped to handle the nonlinearity it induces.

Main contributions. In this work, we answer this question by revisiting classical sparse coding. We show that under superposition, even labels that are linearly separable in latent space may become nonlinearly separable in activation space. We demonstrate that this is particularly pronounced in OOD settings, so that a perfect linear probe trained in-distribution will fail OOD (Figure 1). SAEs perform nonlinear inference, but amortizing it into a fixed encoder introduces a systematic amortization gap Figure 5: the encoder fits the training distribution's co-occurrence structure and fails to generalize to novel combinations of latent factors under OOD composition shift (Figure 3). Classical sparse coding instead solves a per-sample optimisation problem for each input that is well-posed and identifiable under compressed sensing theory, making it robust to compositional shifts by construction (Section 4). Per-sample inference requires a dictionary, which can be learned through classical alternating optimization or reused from an already-trained SAE. The key insight is that per-sample inference at evaluation time is what closes the gap, even when the dictionary was learned by amortised methods. Hybrid approaches that partially undo amortization —running per-sample FISTA with an SAE-learned dictionary, or using the SAE encoder's output as a warm start for iterative refinement—improve OOD performance, confirming that the inference procedure, not the dictionary alone, is the primary bottleneck (Figure 6). However, these hybrid approaches remain below fully per-sample methods with independently learned dictionaries, suggesting that both dictionary quality and initialization bias contribute to the residual gap.

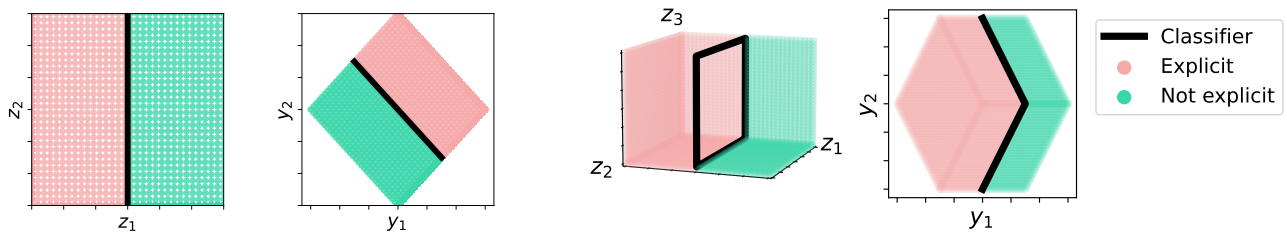
2 RELATED WORK

Compositional generalization—the ability to understand and produce novel combinations of learned concepts—remains a fundamental challenge for neural networks (Fodor and Pylyshyn, 1988; Hupkes et al., 2020). Current approaches in causal representation learning attempt to achieve this through structural constraints, such as *additive* decoders (Lachapelle et al., 2023) (e.g., in SAEs), or specific training objectives like compositional risk minimization (Mahajan et al., 2025). These works focus on obtaining guarantees for the compositional generalization of disentangled models, while here, we evaluate the effect of compositional shifts under superposition.

SAEs have recently emerged as a primary tool for decomposing the internal activations of LLMs into interpretable, monosemantic features (Cunningham et al., 2023). Despite their success in interpretability, their out-of-distribution (OOD) robustness is a growing concern. Recent evaluations suggest that SAEs trained on general datasets often fail to discover generalizable concepts across different domains or layers (Heindrich et al., 2025), underperform compared to simple linear probes (Kantamneni et al., 2025), and remain brittle even when scaled (Gao et al., 2024). Interestingly, this brittleness is less pronounced in domain-specific applications, such as medical QA or pathology, where SAEs have shown more stable and biologically relevant feature transfer (O'Neill et al., 2025; Le et al., 2024). These conflicting results motivate a more principled evaluation of SAEs under distribution shifts. (Joshi et al., 2025).

Recent work has highlighted the limitations of SAEs in recovering true latent variables (O'Neill et al., 2024; Paulo and Belrose, 2025). This stands in contrast to the classical sparse coding framework (Olshausen and Field, 1996; Ranzato et al., 2007), which utilizes iterative optimization rather than a learned encoder to recover latent variables. This iterative approach provides stronger theoretical guarantees for the unique recovery of latents (Hillar and Sommer, 2015; Lewicki and Sejnowski, 1997; Gribonval et al., 2015). In contrast to O'Neill et al. (2024), who explore different in-distribution (ID) amortization strategies, we focus our analysis on downstream tasks under OOD compositional shifts.

Lastly, literature on overcomplete independent component analysis (Podosinnikova et al., 2019; Wang and Seigal, 2024) explores identifiability where the number of latent variables exceeds the number of observed variables ($d_z > d_y$), though these models typically rely on statistical independence rather than sparsity.



(a) (b)

Figure 1: Binary classification with $t = z_1 > 0.5$ (blue: safe, red: unsafe). **(a)** When $d_z = d_y$, the linear decision boundary in latent space remains linear after mixing $\mathbf{y} = \mathbf{W}\mathbf{z}$. **(b)** When $d_z > d_y$ (overcompleteness) and z sparse ($\|.\|_0 \leq 2$), we can project down into non-overlapping regions (i.e., compressed sensing is possible), but the decision boundary becomes nonlinear in activation space, making linear probes insufficient.

3 AMORTISATION VS POINTWISE SPARSE INFERENCE AND COMPOSITIONAL GENERALIZATION

We begin by specifying the data-generating process. Consider latent variable vectors $\mathbf{z} \in \mathbb{R}^n$ with at most k non-zeros. A support set $S \subseteq [n]$ is drawn first to index these non-zero components following the process:

$$\begin{aligned} S \sim p_S, \quad \mathbf{z} \sim p(\mathbf{z} \mid S) \\ \text{supp}(p_S) \subseteq \mathcal{S}_k := \{S \subseteq [n] : |S| \leq k\}, \\ \mathbb{P}(\mathbf{z}_{S^c} = \mathbf{0} \mid S) = 1. \end{aligned} \tag{1}$$

An unknown generative process g maps latents to data, producing $\mathbf{x} := g(\mathbf{z}) \in \mathbb{R}^{d_x}$. Although $\mathbf{z}$ is unobserved, we have access to learned representations $\mathbf{y} := f(\mathbf{x}) \in \mathbb{R}^{d_y}$ for some fixed (encoding) function $f : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$ (e.g., an LLM activation map). Since the coordinates of $\mathbf{y}$ need not necessarily align with those of $\mathbf{z}$, we fit a representation model $r : \mathbb{R}^{d_y} \to \mathbb{R}^{d_h}$ estimating $\mathbf{h} := r(\mathbf{y})$ and another decoder $q : \mathbb{R}^{d_h} \to \mathbb{R}^{d_y}$ s.t. $\hat{\mathbf{y}} := q(\mathbf{h})$. Ideally, $\mathbf{h}$ serves as a proxy for $\mathbf{z}$ and has the same number of components $d_h = n$, while $d_y < n$. In this setting, we refer to individual coordinates of $\mathbf{h}$ (one-dimensional subspaces of $\mathbb{R}^{d_h}$) as *features*, and we seek features that correspond (approximately) to the latent coordinates $\mathbf{z}$.

Typically, $\mathbf{h}$ is enforced to be sparse, in line with Equation (1). In practice, SAEs (Cunningham et al., 2023) are commonly used to learn the autoencoder $q \circ r$, where typically the decoder q is assumed to be linear and the encoder r is a single-layer Perceptron. The assumption of a linear decoder and the mapping to a higher-dimensional $\mathbf{h}$ are motivated by the LRH stating that the composed map $f \circ g$ is linear in the underlying latent variables s.t.,

$$\mathbf{y} = f(g(\mathbf{z})) \approx \mathbf{W}\mathbf{z} + \mathbf{b}, \qquad \mathbf{x} \sim p_{\mathbf{x}}, \tag{2}$$

for some matrix $\mathbf{W} \in \mathbb{R}^{d_y \times d_z}$ and offset $\mathbf{b} \in \mathbb{R}^{d_y}$, where $p_{\mathbf{x}}$ denotes the induced data distribution under the generative process $\mathbf{x} = g(\mathbf{z})$. A long line of work provides evidence for this hypothesis (c.f. Rumelhart and Abrahamson (1973); Hinton et al. (1986); Mikolov et al. (2013); Ravfogel et al. (2020), reviewed in Klindt et al. (2025)). More recently, theoretical work justifies why linear properties could arise in these models (c.f. Jiang et al. (2024); Roeder et al. (2021); Marconato et al. (2024); Reizinger et al. (2024)).

Overcompleteness implies non-injectivity. SAEs are often trained with an overcomplete feature dimension $d_h > d_y$, so that the decoder operates as a linear dictionary $\mathbf{W} \in \mathbb{R}^{d_h \times d_z}$. When $d_h > d_y$, rank–nullity implies that $\mathbf{W}$ cannot be injective: since $\text{rank}(\mathbf{W}) \leq d_y$, we have $\dim \ker(\mathbf{W}) = d_h - \text{rank}(\mathbf{W}) \geq d_h - d_y > 0$ and hence $\ker(\mathbf{W}) \neq \{\mathbf{0}\}$. As a result, the equation $\mathbf{y} = \mathbf{W}\mathbf{h}$ is underdetermined, and each $\mathbf{y}$ admits infinitely many codes $\mathbf{h}$ that reconstruct it. That means the model $\mathbf{y} = \mathbf{W}\mathbf{z}$ is inherently not identifiably since, for any invertible (d_z by d_z) matrix $\mathbf{M}$, we can write $\mathbf{y} = (\mathbf{W}\mathbf{M})(\mathbf{M}^{-1}\mathbf{z}) = \mathbf{W}'\mathbf{z}'$, defining a different dictionary and latent variable that result in the same observed variable $\mathbf{y}$. Thus, interpreting $\mathbf{h} = r(\mathbf{y})$ as recovering latents requires an additional selection principle that prefers one solution among many consistent codes.

To obtain a unique solution, one typically restricts the code to be sparse, mirroring the latent sparsity assumption in Equation (1). Concretely, we restrict $\mathbf{h} \in \mathbb{R}^{d_h}$ to lie in the k-sparse set [1]:

$$\Sigma_k^{(d_h)} := \{\mathbf{h} \in \mathbb{R}^{d_h} : \|\mathbf{h}\|_0 \leq k\},$$

This sparsity can be enforced softly (e.g., via an ℓ_1 penalty) or as a hard constraint as in top-k SAEs, which retain only the k largest-magnitude coordinates of $\mathbf{h}$ per input. Algebraically, sparsity replaces the unconstrained feasibility set $\{\mathbf{h} \in \mathbb{R}^{d_h} : \mathbf{W}\mathbf{h} = \mathbf{y}\}$ (typically infinite) with the constrained set $\{\mathbf{h} \in \Sigma_k^{(d_h)} : \mathbf{W}\mathbf{h} = \mathbf{y}\}$, which can be a singleton under suitable conditions on $\mathbf{W}$. In this sense, $r(\mathbf{y})$ is meaningful only

[1] equivalently requiring that $\text{supp}(\mathbf{h}) \in \mathcal{S}_k^{(d_h)} := \{S \subseteq [d_h] : |S| \leq k\}$.

insofar as it implements a consistent sparse selection of dictionary atoms among the many codes that reconstruct the same $\mathbf{y}$.

Restricting the codes to be sparse is not sufficient to guarantee uniqueness—there may still exist distinct $\mathbf{h}, \mathbf{h}' \in \Sigma_k^{(d_h)}$ with $\mathbf{W}\mathbf{h} = \mathbf{W}\mathbf{h}'$. We need a property of the dictionary $\mathbf{W}$ ensuring that it does not collapse sparse codes and that they are identifiable. A standard sufficient condition is the restricted isometry property (RIP) Candes and Tao (2006); Donoho (2006b): $\mathbf{W}$ satisfies RIP of order s with constant δ_s if $(1-\delta_s)\|\mathbf{h}\|_2^2 \le \|\mathbf{W}\mathbf{h}\|_2^2 \le (1+\delta_s)\|\mathbf{h}\|_2^2 \quad \forall \|\mathbf{h}\|_0 \le s$. That is, the squared norm of $\mathbf{W}\mathbf{h}$ is bounded within a tight range from the squared norm of $\mathbf{h}$. In particular, RIP at order $2k$ implies that $\mathbf{W}$ is injective on $\Sigma_k^{(d_h)}$: if $\mathbf{W}\mathbf{h} = \mathbf{W}\mathbf{h}'$ with $\mathbf{h}, \mathbf{h}' \in \Sigma_k^{(d_h)}$, then $\mathbf{h} - \mathbf{h}' \in \Sigma_{2k}^{(d_h)}$ and RIP forces $\mathbf{h} = \mathbf{h}'$. [2] It is impossible for two different sparse vectors to map to the same output since their difference cannot be in the null space of $\mathbf{W}$ due to RIP, ensuring injectivity.

Sparse Coding and Amortization. Such restricted invertibility conditions are central in *sparse coding* and dictionary learning, where one infers a sparse code per input. Concretely, given samples $\{\mathbf{y}_i\}_{i=1}^p$ and fixed dictionary $\mathbf{W}$, the canonical *pointwise* formulation solves, for each i,

$$\mathbf{h}_i^\star \in \arg\min_{\mathbf{h}\in\mathbb{R}^{d_h}} \tfrac{1}{2}\|\mathbf{y}_i - \mathbf{W}\mathbf{h}\|_2^2 + \lambda\|\mathbf{h}\|_1,$$

(or equivalently, a hard sparsity constraint $\mathbf{h} \in \Sigma_k^{(d_h)}$). Algorithms such as Iterative Shrinkage-Thresholding Algorithm (ISTA) and FISTA compute $\mathbf{h}_i^\star$ by iterating a sequence of code estimates $\mathbf{h}_i^{(0)}, \mathbf{h}_i^{(1)}, \ldots$ for each input, typically starting from $\mathbf{h}_i^{(0)} = \mathbf{0}$ and applying repeated gradient-and-thresholding updates until convergence (Beck and Teboulle, 2009; Daubechies et al., 2004). Under RIP-type assumptions, there exist uniqueness and recovery guarantees for per-input decoding (Candès and Wakin, 2008; Donoho, 2006a). RIP is fulfilled for random Gaussian matrices projecting down into $d_y \ge O(k \ln(\frac{d_h}{k}))$ dimensions.

When the dictionary is unknown, (blind) compressed sensing is paired with *dictionary learning* (Olshausen and Field, 1996; Mairal et al., 2010), which alternates between (i) pointwise inference of $\{\mathbf{h}_i\}_{i=1}^p$ given the current $\mathbf{W}$ and (ii) updating $\mathbf{W}$ to minimise reconstruction error under a column-norm constraint. A standard formulation is,

$$\min_{\mathbf{W},\{\mathbf{h}_i\}_{i=1}^p} \sum_{i=1}^p \left(\tfrac{1}{2}\|\mathbf{y}_i - \mathbf{W}\mathbf{h}_i\|_2^2 + \lambda\|\mathbf{h}_i\|_1\right) \quad \text{s.t.} \|\mathbf{W}_{:,j}\|_2 \le 1 \;\; \forall j \in [d_h]. \tag{3}$$

[2] A classical alternative is $\mathrm{spark}(\mathbf{W}) > 2k$, but spark is typically intractable to compute. If $\mathrm{spark}(\mathbf{W}) > 2k$, then $\ker(\mathbf{W}) \cap \Sigma_{2k}^{(d_h)} = \{\mathbf{0}\}$ and hence $\mathbf{W}$ is injective on $\Sigma_k^{(d_h)}$ (Donoho and Elad, 2003; Gribonval and Nielsen, 2004).

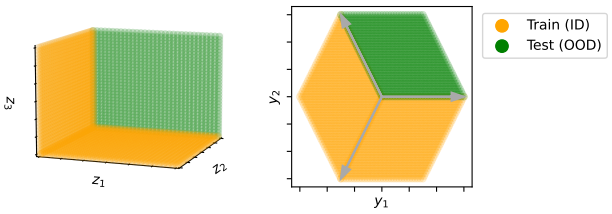


Figure 2: Compositional OOD split. *Left:* In-distribution (ID) training data covers support pairs (z_1, z_2) and (z_2, z_3) and the novel combination (z_1, z_3) is held out for OOD evaluation. *Right:* Same split in activation space $\mathbf{y} = \mathbf{W}\mathbf{z}$.

In contrast, amortized methods replace per-input iterative solving with a feed-forward predictor (Vafaii et al., 2024, 2025). LISTA instantiates this idea by unrolling a fixed number of ISTA iterations and learning their parameters, to approximate $\mathbf{h}_i^\star$ for a given dictionary (Gregor and LeCun, 2010). Crucially, LISTA is *amortized*, i.e. designed as a learned inference procedure for the sparse-coding objective with $\mathbf{W}$ treated as fixed (or tied to the unrolled weights). SAEs also amortize sparse inference, but typically do so while *jointly learning* both the encoder and the dictionary (decoder q) via a dataset-level autoencoding objective; the learned encoder does not need to correspond to an unrolled sparse-coding solver.

On the other hand, typical SAEs with linear-nonlinear encoders have been shown not to be identifiable (O'Neill et al., 2024), suggesting that the encoder does not have enough capacity to invert the non-injective mixing function. We hypothesize that this is one reason for the poor generalization of SAEs OOD found in the literature (Kantamneni et al., 2025). In principle, a disentangled generalization should allow for better OOD generalization on downstream tasks (Schölkopf* et al., 2021). In particular, we study the case where the OOD shift is compositional. With this setup, arrive at two concrete directions:

i What the LRH does (and does not) imply about the *linear separability* of latent variables.

ii How point-wise versus amortized sparse inference affects *compositional generalization*.

Compositional generalization. We evaluate inferred sparse codes via a downstream binary prediction task. E.g., we observe $\mathbf{y}$ and infer $\mathbf{h}$. The target t depends on the latent $\mathbf{z}$ through $t = u(\mathbf{z}) \in \{0, 1\}$; hence, predicting t from $\mathbf{h}$ through $\hat{t} = v(\mathbf{h})$ is a standard supervised proxy for testing whether $\mathbf{h}$ preserves task-relevant information about $\mathbf{z}$, such as by fitting a logistic head on $\mathbf{h}$ and evaluating it through the log-odds $\log \frac{\Pr(t=1|\mathbf{h})}{\Pr(t=0|\mathbf{h})} = \mathbf{a}^\top\mathbf{h} + a_0$ for weights $\mathbf{a}$.

Train and test sets differ in which combinations of generative factors co-occur (Figure 2). In terms of supports $S \subseteq [n]$ of

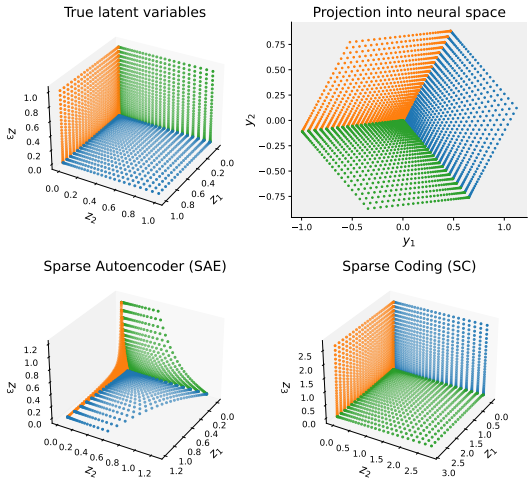


Figure 3: SAEs fail to recover latent variables under superposition, but sparse coding succeeds. **Top left:** Ground-truth latents ($d_z = 3$, $k = 2$); colors denote active-variable combinations. **Top right:** Activation space $\mathbf{y} = \mathbf{W}\mathbf{z}$ ($d_y = 2$); factors overlap after projection. **Bottom left:** SAE reconstruction; planes are not recovered. **Bottom right:** Sparse coding reconstruction; latents are identified up to scaling.

the sparse latent $\mathbf{z}$ (cf. Equation (1)), the ID data excludes a structured subset of support patterns while the OOD data concentrates on those withheld combinations. E.g., consider $k = 2$, and three factors $z_1 = \texttt{subreddit}$, $z_2 = \texttt{tone}$ (e.g. academic, angry, provocative, sensual), and $z_3 = \texttt{language}$ (e.g. Hindi, Portuguese, Inuktut). Suppose the ID data contains $\{\texttt{subreddit}, \texttt{tone}\}$ and $\{\texttt{tone}, \texttt{language}\}$ but withholds $\{\texttt{subreddit}, \texttt{language}\}$. Consider a label $t = \{\texttt{explicit}, \texttt{not explicit}\}$ (e.g. NSFW, SFW) that depends on the variable $\texttt{subreddit}$ (e.g., $t = \mathbf{1}\{z_1 \neq 0\}$). Despite having high ID accuracy, a model may rely on a shortcut where the predictor $\mathbf{a}^\top\mathbf{h}$ tracks coordinates of $\mathbf{h}$ that are merely correlated with $\texttt{subreddit}$ in the training set, including coordinates that encode $\texttt{tone}$. The OOD split breaks these correlations by presenting $\texttt{subreddit}$ in a new context, $\texttt{language}$. Achieving OOD success, therefore, requires the encoder r to recover coordinates where the evidence for t remains stable across different factor recombinations. This follows the withheld-co-occurrence logic central to spurious-correlation benchmarks like Waterbirds (Sagawa et al., 2019).

Compositional OOD is a targeted stress test for sparse-code identifiability *as realised by the inference procedure*: an amortised encoder can mis-select supports under $p_{\text{OOD}}(S)$ because it has internalised training-time co-occurrence structure, whereas per-sample inference recomputes a code guaranteed to be unique under RIP. Figure 3 confirms this: SAEs fail to recover the OOD plane (z_1, z_3), while sparse coding reconstructs all latents.

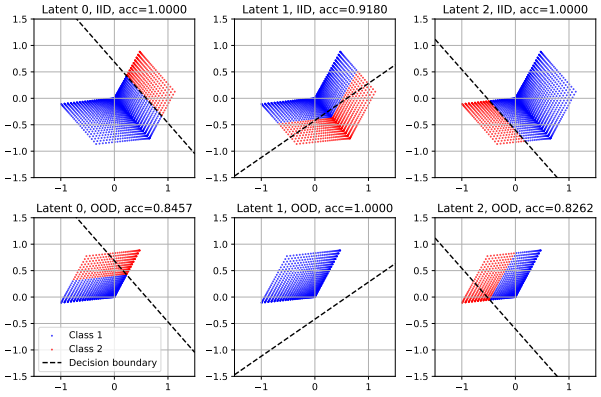


Figure 4: **Linear probes fail OOD under overcompleteness.** Each column sets $t = z_i$. The linear classifier fits the ID decision boundary well, but the compression $\mathbf{y} = \mathbf{A}\mathbf{z}$ introduces nonlinearity that is only exposed OOD, causing catastrophic generalisation failure (columns 1, 3) or, even, poor ID accuracy (column 2).

Linear Probing under the LRH. When $\mathbf{W}$ is injective ($d_z \leq d_y$), linear separability is invariant: $\mathbf{W}$ can rotate or rescale latent space, but a label that is linear in $\mathbf{z}$ remains linear in $\mathbf{y} = \mathbf{W}\mathbf{z}$ (Figure 1a; also (Garg et al., 2026) Fig. 1).

However, under overcompleteness, even labels that are linearly separable in latent space can become nonlinearly separable in activation space. When $\mathbf{W}$ is non-injective, multiple $\mathbf{z}$ map to the same $\mathbf{y}$, potentially lying on opposite sides of the latent hyperplane; in that case, no *linear* rule in $\mathbf{y}$ can match the latent separator without error. This is the geometric phenomenon illustrated in Figure 1b: a hyperplane separator in the full latent space can appear linear on the in-distribution slice of observed mixtures, yet become effectively nonlinear (or even ill-posed [3]) in activation space after being transformed through $\mathbf{W}$. Figure 4 illustrates different failure cases of linear probes under this compositional setting. Under sparse coding, RIP does not make $\mathbf{W}$ globally invertible, but it ensures that $\mathbf{W}$ does not collapse *sparse* directions, so that sparse latents remain (nonlinearly) distinguishable and pointwise sparse inference is well-posed.

4 EXPERIMENTS

We evaluate whether per-sample sparse inference recovers latent factors more robustly than amortised inference under compositional distribution shift. We organise methods into four families that span a spectrum from fully per-sample to fully amortised inference, plus a linear-probe baseline that operates directly in activation space. All methods are evaluated on both in-distribution (ID) and out-of-distribution (OOD) test sets, where the OOD split withholds specific

[3] Ill-posedness arises when $\exists\, \mathbf{z} \neq \mathbf{z}'$ with $\mathbf{W}\mathbf{z} = \mathbf{W}\mathbf{z}'$ but $t(\mathbf{z}) \neq t(\mathbf{z}')$, in which case t is not a function of $\mathbf{y}$ at all.

Table 1: Key ratios. The bound requires $\delta \geq C\,\rho \ln(1/\rho)$ for a constant $C > 0$: sparser codes (smaller ρ) and higher observation dimension (larger δ) make recovery easier.

Symbol	Definition	Interpretation
ρ	$\frac{k}{d_h}$	**Sparsity.** Fraction of non-zeros in $\mathbf{h}$.
δ	$\frac{d_y}{d_h}$	**Undersampling.** Ratio of observation dimension d_y to code dimension d_h. $\delta < 1$ is the overcomplete regime.

combinations of active latents during training (Figure 2).

We consider latent variables $\mathbf{z} \in [0,1]^{d_z}$ with at most k non-zero entries, each sampled uniformly on $[0,1]$ when active, observed through a linear mixing $\mathbf{y} = \mathbf{A}\mathbf{z}$, where $\mathbf{A} \in \mathbb{R}^{d_y \times d_z}$ with $d_y < d_z$. Details in Section A. Two ratios govern recovery difficulty (Table 1): sparsity $\rho = k/d_z$ and undersampling $\delta = d_y/d_z$. We evaluate these using the metrics below. The target $t = \mathbf{1}\{z_1 > 0.5\}$ is predicted from inferred codes $\mathbf{h}$.

Metrics. We report three quantities on both ID and OOD data. mean correlation coefficient (*MCC*) (Hyvarinen and Morioka, 2016) evaluates identifiability of the learned representation $\mathbf{h}$ against the ground truth latent factors $\mathbf{z}$. It should be 1 for a representation identified up to permutation and rescaling. *Accuracy* denotes the logistic probe classification accuracy.

Methods. We compare 17 methods spanning four families (Table 2): *sparse coding* (per-sample ℓ_1 inference with oracle or learned dictionaries), *SAEs* (amortised feedforward encoder), *frozen decoder* (per-sample FISTA on a frozen SAE-learned dictionary), and *refined hybrids* (FISTA warm-started from SAE codes). The last two families disentangle dictionary quality from inference: frozen-decoder methods replace only the encoder while reusing the SAE's dictionary; refined methods additionally test whether the SAE's output provides a useful initialisation. A linear-probe baseline operating directly on $\mathbf{y}$ completes the comparison. Architectural and optimisation details are in C.1. All SAEs (ReLU (Cunningham et al., 2023), JumpReLU (Rajamanoharan et al., 2024), Top-K (Gao et al., 2024), MP (Costa et al., 2025)) share the same decoder dimension and are trained with identical optimiser settings; they differ only in the activation function governing the encoder's sparsity mechanism.

4.1 THE AMORTISATION GAP PERSISTS ACROSS UNDERSAMPLING RATIOS

Compressed sensing theory predicts a phase transition in sparse recovery: once the number of observations d_y exceeds $O(k \ln(d_z/k))$, per-sample ℓ_1 methods recover the latent code exactly. We sweep the undersampling ratio $\delta = d_y/d_z$

Table 2: **Legends for all plots.** S = supervised (oracle $\mathbf{W}$); all U = unsupervised (learned).

Method	Inference	Dictionary
Sparse Coding — per-sample ℓ_1 recovery		
● FISTA$_S$	FISTA	oracle $\mathbf{W}$
◆ LISTA$_S$	unrolled ISTA	oracle $\mathbf{W}$
■ DL-FISTA$_U$	FISTA	learned
▲ SOFTPLUS$_U$	diff. relaxation	learned
SAE — amortised encoder		
● SAE(RELU)$_U$	feedforward	learned
■ SAE(TOPK)$_U$	feedforward	learned
▲ SAE(JUMPRELU)$_U$	feedforward	learned
◆ SAE(MP)$_U$	feedforward	learned
Frozen Decoder — FISTA on frozen SAE decoder $\hat{\mathbf{W}}$		
● FISTA+SAE(RELU)$_U$	FISTA	frozen SAE
■ FISTA+SAE(TOPK)$_U$	FISTA	frozen SAE
▲ FISTA+SAE(JUMPRELU)$_U$	FISTA	frozen SAE
◆ FISTA+SAE(MP)$_U$	FISTA	frozen SAE
Refined — FISTA warm-started from SAE codes		
● REFINED(RELU)$_U$	FISTA (warm)	frozen SAE
■ REFINED(TOPK)$_U$	FISTA (warm)	frozen SAE
▲ REFINED(JUMPRELU)$_U$	FISTA (warm)	frozen SAE
◆ REFINED(MP)$_U$	FISTA (warm)	frozen SAE
Baseline		
▪ LINEAR PROBE$_S$	logistic regression on $\mathbf{y}$	

across a grid of latent dimensions $d_z \in \{50, 100, 200\}$ and sparsity levels $k \in \{3, 5, 10\}$, and report ID MCC (Figure 5). Phase transition with other metrics are reported in Section D. If SAEs solved the same sparse inference problem, they would exhibit the same transition and reach the same asymptotic performance. A persistent gap between the two would reveal the cost of amortising inference into a fixed encoder.

Per-sample methods exhibit the predicted phase transition. FISTA$_S$ (oracle dictionary) transitions sharply to near-perfect MCC once δ passes a critical threshold and the transition sharpens with d_z, matching the theoretical prediction. DL-FISTA$_U$ follows the same curve, shifted right by the cost of learning the dictionary. SAE variants also improve with δ—they are not insensitive to the undersampling ratio—but plateau at 0.2–0.5 MCC, well below the near-perfect recovery that per-sample methods achieve in the same regime. Crucially, the gap does not close at high δ: even when the problem is well within the regime where compressed sensing guarantees exact recovery, the amortised encoder remains the bottleneck.

Takeaway. Both method families benefit from less aggressive undersampling (higher δ), but SAEs saturate far below per-sample methods.

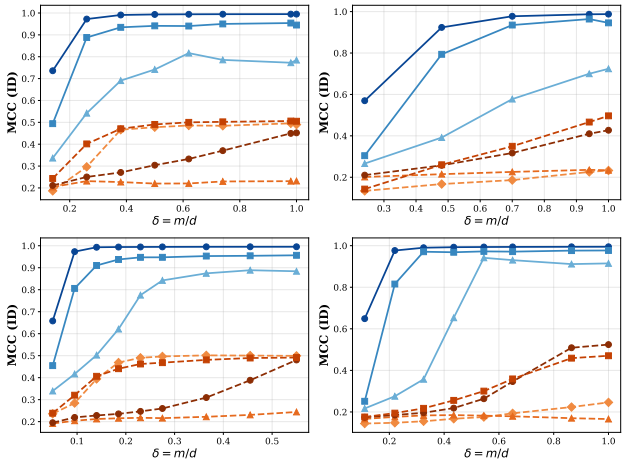


Figure 5: **The amortisation gap persists across undersampling ratios.** *Clockwise from top*: ($n = 50, k = 3$), ($n = 50, k = 10$), ($n = 200, k = 3$), ($n = 50, k = 10$). DL-FISTA$_U$ (unsupervised sparse coding) obtains a significantly higher MCC than all the SAEs (legend in Tab. 2) in all sweeps across undersampling ratios.

4.2 SCALING UP LATENT DIMENSION DOES NOT CLOSE THE COMPOSITIONAL GAP

All unsupervised methods face a more challenging problem as d_z grows, as the dictionary has more columns and the space of sparse patterns expands combinatorially. Per-sample methods should degrade more gracefully if the bottleneck is dictionary learning; SAEs should degrade faster if the bottleneck is amortised inference.

We sweep $d_z \in \{50, 100, 500, 1\text{K}, 5\text{K}, 10\text{K}\}$ with k and δ held fixed and report ID MCC and OOD AUC (Figure 6, left column). FISTA$_S$ is the only method that remains near-perfect across the full range (MCC ≥ 0.95, OOD AUC ≈ 1.0). DL-FISTA$_U$ degrades substantially in absolute MCC as d_z grows—dropping to ~0.3 by $d_z = 10\text{K}$—but this reflects the difficulty of dictionary learning, not a compositional failure: its OOD AUC tracks its ID performance, maintaining a small ID–OOD gap throughout. SAE variants sit at comparable or lower absolute MCC, but critically exhibit a *large and persistent ID–OOD gap*: OOD AUC clusters between 0.5 and 0.8 with high variance, indicating that codes which appear informative in-distribution fail to generalise to novel support patterns. Additionally, the linear probe degrades sharply as superposition intensifies, confirming the limitations of linear decoding.

Varying sparsity k with d_z fixed yields an analogous pattern: per-sample methods degrade gracefully while SAE OOD AUC converges to chance (~0.5) at high k (Figure 10).

Takeaway. Per-sample methods degrade in absolute performance as d_z grows, but do so uniformly across ID and OOD. SAEs exhibit a persistent ID–OOD gap, revealing a compositional generalisation failure distinct from the difficulty of dictionary learning.

4.3 MORE DATA DOES NOT CLOSE THE AMORTISATION GAP

One issue with the evaluation setup could be that SAEs are simply data-limited: with enough training samples, the encoder should learn to approximate per-sample inference. If this were the case, the gap between SAEs and sparse coding would shrink as the training set grows.

We vary the number of training samples $p \in \{10^2, \ldots, 10^5\}$ with all other parameters held fixed and report ID MCC and OOD AUC (Figure 7). FISTA$_S$ is constant by construction (it uses no training data beyond the oracle dictionary). DL-FISTA$_U$ benefits substantially from more data—its MCC jumps from ~0.5 at $p = 10^2$ to ~0.98 by $p = 10^3$ and saturates—confirming that dictionary learning is genuinely sample-limited and that per-sample inference exploits a better dictionary immediately. SAE variants show a different pattern. SAE(TOPK) and SAE(RELU) improve only marginally, plateauing around 0.35–0.45 MCC. SAE(JUMPRELU) *degrades* with more data, dropping from ~0.45 to below 0.1, suggesting that additional training leads to a poor local solution rather than correcting it. The gap between per-sample and amortised methods is stable or widening across two orders of magnitude of additional data. On OOD AUC, per-sample methods converge to ~1.0 while SAEs remain scattered between 0.5 and 0.85 with high variance, showing no trend toward closing the gap.

Takeaway. Additional training data closes the dictionary learning gap (benefiting DL-FISTA$_U$) but does not close the amortisation gap. This points to issues with the encoder architecture, not sample complexity.

4.4 DISENTANGLING DICTIONARY QUALITY FROM INFERENCE

The preceding experiments confound two potential sources of SAE failure: a poor dictionary and poor inference. The frozen-decoder and refined families isolate these by holding the dictionary fixed and varying only the inference procedure. For each SAE variant we compare three conditions: the raw SAE (circle), FISTA run on the frozen SAE decoder initialised from $\mathbf{h}^{(0)} = \mathbf{0}$ (frozen decoder, triangle pointing up), and FISTA warm-started from the SAE encoder's output (refined, triangle pointing down). Figure 6 (middle and right columns) shows per-variant lollipop plots of the improvement on ID MCC and OOD AUC as d_z varies; Figure 8 in the appendix reports all six metric panels.

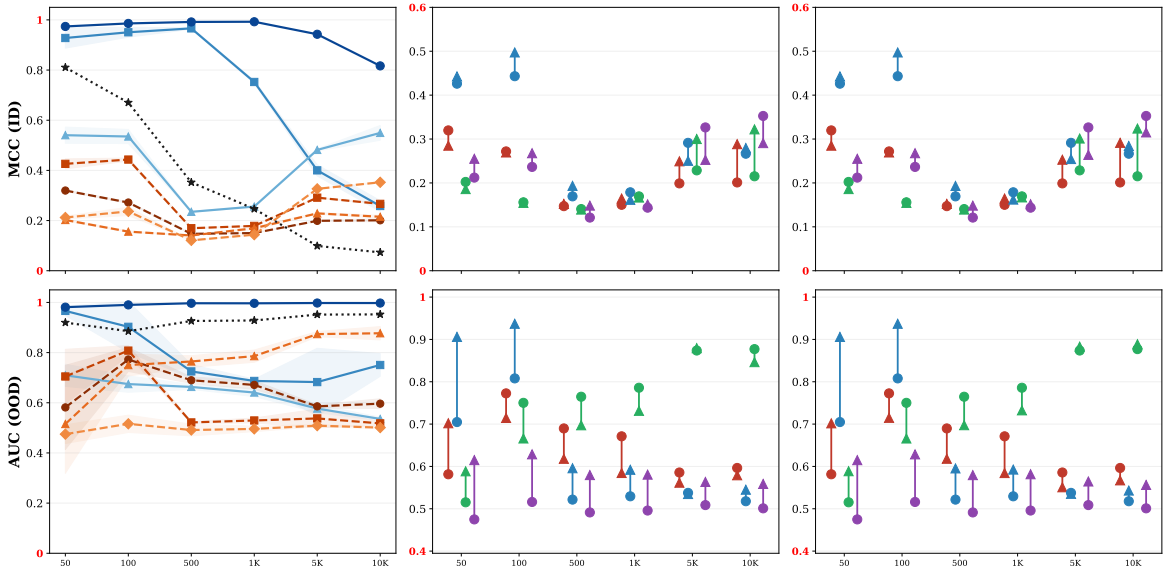


Figure 6: **Horizontal axis: number of latent variables.** *Left*: Scaling up the latent dimension does not close the compositional gap. FISTA$_S$ outperforms linear probes on MCC and AUC in all settings as superposition intensifies. *Middle and Right*: Disentangling dictionary quality from inference. SAEs gain by being combined with FISTA (hybrid approach). Sweeps latents with $k = 10$, number of samples = 5000, input dimension d_y follows the compressed sensing bound with constant 2.

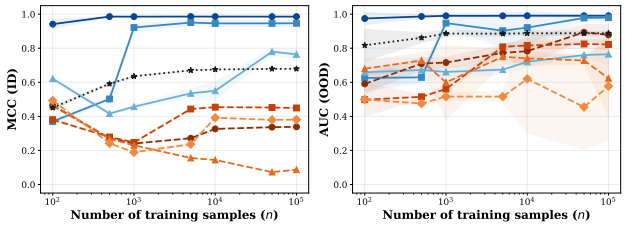


Figure 7: **More data does not close the amortisation gap.** $d_h = 100, k = 10, d_y = 47$. (Left) DL-FISTA$_U$'s MCC and AUC increase as the number of samples increases, outperforming all the other models (besides its supervised counterpart).

Implications. (i) Replacing the SAE encoder with per-sample FISTA while reusing the SAE's dictionary consistently improves both MCC and OOD AUC across all four SAE variants and all d_z. The gains are largest on OOD AUC, confirming that a substantial portion of the SAE's OOD failure is due to the encoder, not the dictionary. The magnitude of improvement varies across activation functions: FISTA+SAE(JUMPRELU) shows the largest gains, suggesting that JumpReLU learns a comparatively better dictionary that is held back by its amortised encoder. (ii) Warm-starting FISTA from the SAE's codes yields improvements comparable to the frozen-decoder variants, suggesting that the benefit comes from running per-sample inference to convergence rather than from the choice of initialisation. *Residual gap.* Both hybrid families improve substantially over raw SAEs but remain below DL-FISTA$_U$ with an independently learned dictionary, indicating that the SAE-learned decoder is itself a weaker dictionary than one learned by the classical alternating-minimisation pipeline.

Takeaway. Per-sample inference at test time recovers a large fraction of the SAE's OOD deficit, even when using the SAE's own dictionary. The amortisation gap is the dominant error source, followed by dictionary quality; both contribute, but swapping the encoder alone yields the largest single improvement.

5 CONCLUSION

This paper investigates the utility of sparse coding for compositional generalization under superposition, particularly in out-of-distribution (OOD) scenarios. We demonstrate that superposition distorts feature geometry, making concepts linearly inseparable and necessitating nonlinear probes for accurate detection. Our theoretical framework, validated through simulations, shows that even a perfectly trained linear probe may exhibit degraded performance OOD due to superposition structure. While SAEs with powerful MLP encoders can learn the nonlinear mapping in-distribution (O'Neill et al., 2024), they fail to generalize OOD, highlighting that true sparse inference cannot be easily amortized (Cremer et al., 2018; Zhang et al., 2022). In contrast, classical sparse coding with iterative inference robustly recovers features both in- and out-of-distribution. Our contributions to this study are twofold: first, we conduct extensive experiments on synthetic datasets. Our results consistently show that sparse coding outperforms both linear probes and SAEs

in OOD tasks, underscoring its effectiveness in handling the complexities introduced by superposition. Second, we provide a theoretical analysis of the limitations of linear probes in the presence of superposition, elucidating the geometric distortions that arise. Overall, this study clarifies the critical distinction between linear representation and linear separability, advocating for methods equipped to handle the nonlinearity induced by superposition.

References

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

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.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, and Amanda Askell. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, page 2, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features.

Emmanuel J. Candes and Terence Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52 (12):5406–5425, 2006. doi: 10.1109/TIT.2006.885507.

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

Sviatoslav Chalnev, Matthew Siu, and Arthur Conmy. Improving steering vectors by targeting sparse autoencoder features. *arXiv preprint arXiv:2411.02193*, 2024.

David Chanin, Tomáš Dulka, and Adrià Garriga-Alonso. Feature hedging: Correlated features break narrow sparse autoencoders. *arXiv preprint arXiv:2505.11756*, 2025.

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

Chris Cremer, Xuechen Li, and David Duvenaud. Inference suboptimality in variational autoencoders. In *International conference on machine learning*, pages 1078–1086. PMLR, 2018.

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

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. doi: 10.1002/cpa.20042.

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

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

David L Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via 1 minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202, 2003.

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, September 2022. URL http://arxiv.org/abs/2209.10652. arXiv:2209.10652 [cs].

Jerry A Fodor and Zenon W Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28 (1-2):3–71, 1988.

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

Nikhil Garg, Jon Kleinberg, and Kenny Peng. How many features can a language model store under the linear representation hypothesis?, 2026. URL https://arxiv.org/abs/2602.11246.

Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th international conference on international conference on machine learning*, pages 399–406, 2010.

Rémi Gribonval and Morten Nielsen. Sparse representations in unions of bases. *IEEE transactions on Information theory*, 49(12):3320–3325, 2004.

Rémi Gribonval, Rodolphe Jenatton, and Francis Bach. Sparse and spurious: dictionary learning with noise and outliers, 2015. URL https://arxiv.org/abs/1407.5155.

Lovis Heindrich, Philip Torr, Fazl Barez, and Veronika Thost. Do sparse autoencoders generalize? a case study of answerability, 2025. URL https://arxiv.org/abs/2502.19964.

Christopher J. Hillar and Friedrich T. Sommer. When can dictionary learning uniquely recover sparse data from subsamples? *IEEE Transactions on Information Theory*, 61 (11):6290–6297, 2015. doi: 10.1109/TIT.2015.2460238.

Geoffrey E Hinton et al. Learning distributed representations of concepts. In *Proceedings of the eighth annual conference of the cognitive science society*, volume 1, page 12. Amherst, MA, 1986.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795, 2020.

Aapo Hyvarinen and Hiroshi Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ica. *Advances in neural information processing systems*, 29, 2016.

Yibo Jiang, Goutham Rajendran, Pradeep Ravikumar, Bryon Aragam, and Victor Veitch. On the origins of linear representations in large language models, 2024. URL https://arxiv.org/abs/2403.03867.

Shruti Joshi, Andrea Dittadi, Sébastien Lachapelle, and Dhanya Sridhar. Identifiable steering via sparse autoencoding of multi-concept shifts, 2025. URL https://arxiv.org/abs/2502.12179.

Subhash Kantamneni, Joshua Engels, Senthooran Rajamanoharan, Max Tegmark, and Neel Nanda. Are sparse autoencoders useful? a case study in sparse probing. *arXiv preprint arXiv:2502.16681*, 2025.

Minyoung Kim and Vladimir Pavlovic. Reducing the amortization gap in variational autoencoders: A bayesian random function approach. *arXiv preprint arXiv:2102.03151*, 2021.

David Klindt, Charles O'Neill, Patrik Reizinger, Harald Maurer, and Nina Miolane. From superposition to sparse codes: interpretable representations in neural networks. *arXiv preprint arXiv:2503.01824*, 2025.

Sébastien Lachapelle, Divyat Mahajan, Ioannis Mitliagkas, and Simon Lacoste-Julien. Additive decoders for latent variables identification and cartesian-product extrapolation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 25112–25150. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/4ef594af0d9a519db8fb292452c461fa-Paper-Conference.pdf.

Nhat Minh Le, Ciyue Shen, Neel Patel, Chintan Shah, Darpan Sanghavi, Blake Martin, Alfred Eng, Daniel Shenker, Harshith Padigela, Raymond Biju, Syed Ashar Javed, Jennifer Hipp, John Abel, Harsha Pokkalla, Sean Grullon, and Dinkar Juyal. Learning biologically relevant features in a pathology foundation model using sparse autoencoders, 2024. URL https://arxiv.org/abs/2407.10785.

Michael Lewicki and Terrence J Sejnowski. Learning nonlinear overcomplete representations for efficient coding. In M. Jordan, M. Kearns, and S. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. MIT Press, 1997. URL https://proceedings.neurips.cc/paper_files/paper/1997/file/489d0396e6826eb0c1e611d82ca8b215-Paper.pdf.

Divyat Mahajan, Mohammad Pezeshki, Charles Arnal, Ioannis Mitliagkas, Kartik Ahuja, and Pascal Vincent. Compositional risk minimization, 2025. URL https://arxiv.org/abs/2410.06303.

Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(1), 2010.

Stéphane G Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 41(12):3397–3415, 1993.

Emanuele Marconato, Sébastien Lachapelle, Sebastian Weichwald, and Luigi Gresele. All or none: Identifiable linear properties of next-token predictors in language modeling. *arXiv preprint arXiv:2410.23501*, 2024.

Charles C Margossian and David M Blei. Amortized variational inference: When and why? *arXiv preprint arXiv:2307.11018*, 2023.

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. URL https://aclanthology.org/N13-1090/.

Andrew Ng et al. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19, 2011.

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

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

Charles O'Neill, Mudith Jayasekara, and Max Kirkby. Resurrecting the salmon: Rethinking mechanistic interpretability with domain-specific sparse autoencoders, 2025. URL https://arxiv.org/abs/2508.09363.

Dylan M Paiton, Charles G Frye, Sheng Y Lundquist, Joel D Bowen, Ryan Zarcone, and Bruno A Olshausen. Selectivity and robustness of sparse coding networks. *Journal of vision*, 20(12):10–10, 2020.

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

Gonçalo Paulo and Nora Belrose. Sparse autoencoders trained on the same data learn different features. *arXiv preprint arXiv:2501.16615*, 2025.

Anastasia Podosinnikova, Amelia Perry, Alexander S. Wein, Francis Bach, Alexandre d'Aspremont, and David Sontag. Overcomplete independent component analysis via sdp. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 2583–2592. PMLR, 16–18 Apr 2019. URL https://proceedings.mlr.press/v89/podosinnikova19a.html.

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

Marc'Aurelio Ranzato, Y-Lan Boureau, and Yann LeCun. Sparse feature learning for deep belief networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 20, 2007. URL https://papers.nips.cc/paper_files/paper/2007/hash/2270a5fc66d369cd6c85026c045563b0-Abstract.html.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. Null it out: Guarding protected attributes by iterative nullspace projection. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7237–7256, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.647. URL https://aclanthology.org/2020.acl-main.647.

Patrik Reizinger, Alice Bizeul, Attila Juhos, Julia E Vogt, Randall Balestriero, Wieland Brendel, and David Klindt. Cross-entropy is all you need to invert the data generating process. *arXiv preprint arXiv:2410.21869*, 2024.

Geoffrey Roeder, Luke Metz, and Durk Kingma. On linear identifiability of learned representations. In *International Conference on Machine Learning*, pages 9030–9039. PMLR, 2021.

David E Rumelhart and Adele A Abrahamson. A model for analogical reasoning. *Cognitive Psychology*, 5(1):1–28, 1973.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

B. Schölkopf*, F. Locatello*, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109 (5):612–634, 2021. doi: 10.1109/JPROC.2021.3058954. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9363924. *equal contribution.

Lukas Schott, Julius Von Kügelgen, Frederik Träuble, Peter Gehler, Chris Russell, Matthias Bethge, Bernhard Schölkopf, Francesco Locatello, and Wieland Brendel. Visual representation learning does not generalize strongly within the same domain. *arXiv preprint arXiv:2107.08221*, 2021.

Lewis Smith. The 'strong' feature hypothesis could be wrong. AI Alignment Forum, August 2024. URL https://www.alignmentforum.org/posts/tojtPCCRpKLSHBdpn/the-strong-feature-hypothesis-could-be-wrong. [Accessed 02-26-2026].

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

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

Hadi Vafaii, Dekel Galor, and Jacob Yates. Poisson variational autoencoder. *Advances in Neural Information Processing Systems*, 37:44871–44906, 2024.

Hadi Vafaii, Dekel Galor, and Jacob L Yates. Brain-like variational inference. *ArXiv*, pages arXiv–2410, 2025.

Kexin Wang and Anna Seigal. Identifiability of overcomplete independent component analysis, 2024. URL https://arxiv.org/abs/2401.14709.

Mingtian Zhang, Peter Hayes, and David Barber. Generalization gap in amortized inference. *Advances in neural information processing systems*, 35:26777–26790, 2022.

A SYNTHETIC DATA DETAILS

We generate the synthetic data as follows. We generate a projection matrix $A \in \mathbb{R}^{m \times n}$ with $m < n$, where the elements of A are drawn from a standard Normal distribution, in agreement with the Restricted Isometry Property from compressed sensing. The rows of A are normalized to have unit norm. We generate the latent variables $z \in [0, 1]^n$ with k non-zero components, where the non-zero components are sampled uniformly from $[0, 1]$. The observed variables are then generated as $y = Az$.

When selecting which combinations of latent variables to be active, which k out of n, we consider the particular "out-of-variable" case for OOD generalizations, where some combinations of the variables are not available in the training data. The number of OOD variables is $n/2$. Then, we consider two possibilities:

- **ID data**: Divided into two cases:
  - The first latent variable is active and the other $k - 1$ are drawn between the variables of indices $[2, n/2]$.
  - The first latent variable is not active. The k active indices are drawn from the fool pool of indices $[2, n]$.
- **OOD data**: The first latent variable is active and the other $k - 1$ are drawn between the variables of indices $[n/2 + 1, n]$.

The training set consists of ID data, and the test set consists of OOD data.

B TRAINING DETAILS

We implement the models in PyTorch and train them on a single NVIDIA A100 GPU.

C SPARSE INFERENCE METHODS FOR INTERPRETABILITY

Sparse autoencoders (SAEs) have become the dominant tool for extracting interpretable features from neural network representations (Bricken et al., 2023; Cunningham et al., 2023). An SAE decomposes an activation $\mathbf{y} \in \mathbb{R}^{d_y}$ as $\mathbf{y} \approx \mathbf{Dh}$, where $\mathbf{D} \in \mathbb{R}^{d_h \times d_y}$ is an overcomplete dictionary ($d_h > d_y$) and $\mathbf{h} \in \mathbb{R}^{d_h}$ is a sparse code whose nonzero entries identify the active features. The quality of the interpretation depends entirely on the quality of $\mathbf{h}$: if the codes are wrong, the resulting feature attribution is wrong, regardless of reconstruction fidelity.

Standard SAEs compute codes in a single feedforward pass, $\mathbf{h} = \sigma(\mathbf{W}^\top(\mathbf{y} - b_{\text{pre}}) + b)$, where σ is ReLU or a top-k or JumpReLU activation. This is an *amortised* approximation to the sparse inference problem

$$\mathbf{h}^* = \arg\min_{\mathbf{h}} \ \tfrac{1}{2}\|\mathbf{y} - \mathbf{Dh}\|_2^2 + \lambda\|\mathbf{h}\|_1 \,, \tag{4}$$

which is the Lasso (Tibshirani, 1996), a convex problem with a unique solution (under mild conditions on $\mathbf{D}$). The amortisation gap $\mathbf{h} - \mathbf{h}^*$ is a structured error that is largest precisely when features are correlated or hierarchically organised (Costa et al., 2025; Chanin et al., 2025)—the regimes most relevant to real neural network representations.

Below we compare three inference strategies that move progressively closer to solving Equation (4): FISTA, LISTA, and Matching Pursuit. The comparison focuses on properties that matter for interpretability rather than reconstruction.

C.1 ALGORITHMS

FISTA (Fast Iterative Shrinkage-Thresholding). FISTA (Beck and Teboulle, 2009) solves Equation (4) by alternating a gradient step on the reconstruction loss with the proximal operator for the ℓ_1 penalty (soft-thresholding S_λ), accelerated by Nesterov momentum. Let $\mathbf{h}^{(t)}$ denote the code estimate at iteration t and $\mathbf{q}^{(t)}$ a momentum-extrapolated point:

$$\mathbf{q}^{(t)} = \mathbf{h}^{(t)} + \frac{t_k - 1}{t_{k+1}}\left(\mathbf{h}^{(t)} - \mathbf{h}^{(t-1)}\right), \tag{5}$$

$$\mathbf{h}^{(t+1)} = S_{\eta\lambda}\left(\mathbf{q}^{(t)} - \eta\,\mathbf{D}^\top(\mathbf{D}\,\mathbf{q}^{(t)} - \mathbf{y})\right), \tag{6}$$

where $\eta \le 1/\|\mathbf{D}^\top\mathbf{D}\|_{\text{op}}$ is the step size. Every iteration updates *all* d_h coefficients simultaneously. The support (which atoms are active) is fluid: a coefficient can be driven to zero by soft-thresholding at step t and revived at step $t' > t$. Convergence to the global optimum is guaranteed at rate $O(1/t^2)$ (Beck and Teboulle, 2009). There are no learned parameters; the algorithm is fully determined by $\mathbf{D}$ and λ.

Practical note. Precomputing $\mathbf{W} = \mathbf{I} - \eta\,\mathbf{D}^\top\mathbf{D}$ and $\mathbf{b} = \eta\,\mathbf{D}^\top\mathbf{y}$ reduces each iteration to $\mathbf{h}^{(t+1)} = S_{\eta\lambda}(\mathbf{W}\,\mathbf{h}^{(t)} + \mathbf{b})$: a single matrix–vector multiply plus elementwise thresholding, both fully batchable on GPU.

LISTA (Learned ISTA). LISTA (Gregor and LeCun, 2010) takes the ISTA update (i.e. Equation (6) without momentum) and untethers its parameters from $\mathbf{D}$. Each layer t computes:

$$\mathbf{h}^{(t+1)} = S_{\theta_t}\big(\mathbf{W}_t\,\mathbf{h}^{(t)} + \mathbf{B}_t\,\mathbf{y}\big)\,, \tag{7}$$

where $\mathbf{W}_t \in \mathbb{R}^{d_h \times d_h}$, $\mathbf{B}_t \in \mathbb{R}^{d_h \times d_y}$, and $\theta_t \in \mathbb{R}^{d_h}$ are *free parameters learned by backpropagation*, independently at each layer. In ISTA, $\mathbf{W}_t = \mathbf{I} - \eta\,\mathbf{D}^\top\mathbf{D}$, $\mathbf{B}_t = \eta\,\mathbf{D}^\top$, and $\theta_t = \eta\lambda\,\mathbf{1}$ for all t; LISTA relaxes these constraints, allowing the network to learn iteration-dependent acceleration. Empirically, LISTA matches FISTA's solution quality in 10–20 layers rather than 100+ iterations (Gregor and LeCun, 2010).

Crucially, LISTA retains the structural properties of ISTA/FISTA: all coefficients are updated jointly at every layer, soft-thresholding provides a continuous sparsity mechanism, and the architecture is fully parallelisable across the batch dimension. The dictionary $\mathbf{D}$ (or its learned analogue in $\mathbf{W}_t, \mathbf{B}_t$) can be trained end-to-end.

MP-SAE (Matching Pursuit SAE). MP-SAE (Costa et al., 2025) unrolls the classical matching pursuit algorithm (Mallat and Zhang, 1993) into a differentiable encoder. Let $\mathbf{d}_j$ denote the j-th column of $\mathbf{D}$. At each step $t = 1, \ldots, T$:

$$j^{(t)} = \underset{j \in \{1,\ldots,d_h\}}{\arg\max}\ \mathbf{d}_j^\top \mathbf{r}^{(t-1)}\,, \tag{8}$$

$$h_{j^{(t)}} = \mathbf{d}_{j^{(t)}}^\top \mathbf{r}^{(t-1)}\,, \tag{9}$$

$$\mathbf{r}^{(t)} = \mathbf{r}^{(t-1)} - h_{j^{(t)}}\,\mathbf{d}_{j^{(t)}}\,, \tag{10}$$

where $\mathbf{r}^{(0)} = \mathbf{y} - b_{\text{pre}}$. One atom is selected per step; its coefficient is computed by projection onto the residual; the residual is updated by subtracting the selected atom's contribution. Previous coefficients are never revised. The dictionary is trained end-to-end via backpropagation through the unrolled steps.

MP-SAE approximately solves a different problem from Equation (4): it targets $\min_{\mathbf{h}} \|\mathbf{y} - \mathbf{D}\,\mathbf{h}\|_2^2$ subject to $\|\mathbf{h}\|_0 \le T$, which is NP-hard; matching pursuit is a greedy approximation with no global optimality guarantee.

C.2 COMPARISON ON INTERPRETABILITY-RELEVANT AXES

Well-posedness of codes. FISTA computes the unique minimiser of the Lasso objective Equation (4). The codes are *defined* by a convex optimisation problem: one can point to the objective and state precisely what the codes mean. LISTA approximates this same solution with learned acceleration. MP-SAE computes the output of a greedy procedure that does not correspond to the global minimum of any fixed objective; the codes depend on the selection order, which is itself a function of the dictionary geometry and the input. For identifiability — where "meaning" is invariance across the equivalence class of valid solutions — the distinction matters: the Lasso solution is unique and characterisable; the MP output is not.

Joint coefficient adjustment. FISTA and LISTA update all d_h coefficients at every iteration. If activating atom i changes the optimal coefficient for atom j (as occurs whenever $\mathbf{d}_i^\top\mathbf{d}_j \neq 0$), subsequent iterations correct for this. MP-SAE sets each coefficient once, at the step the atom is selected, and never revises it. Consider $\mathbf{y} = \alpha_1\mathbf{d}_1 + \alpha_2\mathbf{d}_2$ with $\mathbf{d}_1^\top\mathbf{d}_2 = \rho > 0$. MP selects $\mathbf{d}_1$ first (assuming $\alpha_1 > \alpha_2$) and assigns $h_1 = \mathbf{d}_1^\top\mathbf{y} = \alpha_1 + \alpha_2\rho$, which is inflated by $\mathbf{d}_2$'s contribution leaking through the correlation. The coefficient h_2 computed on the residual is correspondingly deflated. FISTA converges to the correct (α_1, α_2) because it jointly adjusts both coefficients across iterations.

Support dynamics. In FISTA/LISTA, the active set (support of $\mathbf{h}$) is fluid: an atom can be activated, deactivated, and reactivated across iterations as the algorithm converges. This self-correction is critical when the initial support estimate is wrong. In MP-SAE, the support grows monotonically — once an atom is selected, it remains active. There is no mechanism to deselect an incorrectly chosen atom, and the error propagates through all subsequent residuals.

Correlated and hierarchical features. Standard SAEs compute all inner products $\langle \mathbf{d}_j, \mathbf{y}\rangle$ simultaneously and threshold, making all activation decisions in parallel. This implicitly assumes quasi-orthogonality of the dictionary (Costa et al., 2025): if $\mathbf{d}_i$ and $\mathbf{d}_j$ are correlated, activating $\mathbf{d}_i$ should reduce the evidence for $\mathbf{d}_j$, but the one-shot encoder cannot express this.

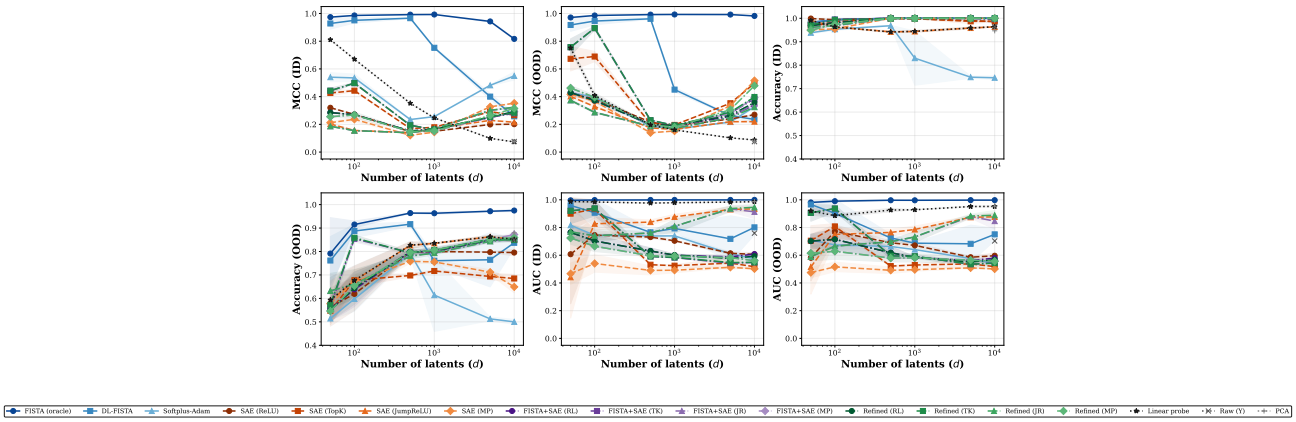


Figure 8: Performance vs number of latent variables.

MP-SAE fixes the conditioning problem via the residual update Equation (10): after selecting $\mathbf{d}_i$, atom $\mathbf{d}_j$ is evaluated against the residual $\mathbf{r}$ rather than the raw input, so correlated atoms no longer double-count shared variance. This is also why MP-SAE naturally recovers hierarchical structure: the first iteration selects the dominant (coarse) feature, and subsequent iterations select progressively finer features on the residual.

FISTA/LISTA handle correlated features correctly *and* with correct magnitudes, because the joint coefficient update avoids the inflation effect described above. However, they do not provide a natural ordering over features — all coefficients converge simultaneously rather than being produced in sequence. When a hierarchy readout is desired, the convergence order or coefficient magnitude in FISTA can serve as a proxy, but the sequential atom selection in MP provides this more directly.

Computational cost. Table 3 summarises the per-step and total cost for a batch of B samples. FISTA and LISTA are fully parallelisable across the batch; MP-SAE's sequential atom selection (the arg max in Equation (8) depends on the previous step's residual) limits GPU utilisation. LISTA compensates for its per-step cost by converging in far fewer steps than FISTA.

Table 3: Computational comparison of sparse inference methods. d_y: input dimension, d_h: dictionary size, T: number of steps/layers, B: batch size.

	FISTA	**LISTA**	**MP-SAE**
Per-step cost	$O(B\,d_h^2)$	$O(B\,d_h^2)$	$O(B\,d_h)$
Typical steps T	100–300	10–20	k (active atoms)
GPU parallelism	Full	Full	Limited
End-to-end trainable	No[4]	Yes	Yes

Trainability. LISTA and MP-SAE are both end-to-end trainable: the dictionary is updated by backpropagation through the unrolled inference steps, using standard deep learning optimisers. FISTA requires alternating optimisation — an outer loop updating $\mathbf{D}$ and an inner loop running FISTA to convergence for each batch — which is slower but provides stronger guarantees on code optimality. A practical middle ground is to train the dictionary using a standard SAE or LISTA, then compute codes at evaluation time using FISTA with the learned dictionary, optionally warm-started from the encoder's output.

D EXPERIMENTAL RESULTS

D.1 SCALING NUMBER OF LATENTS

Figure 8.

[4]FISTA itself is not trained; the dictionary is updated in a separate alternating minimisation step. However, FISTA can be used at evaluation time with a dictionary trained by any method, including an SAE.

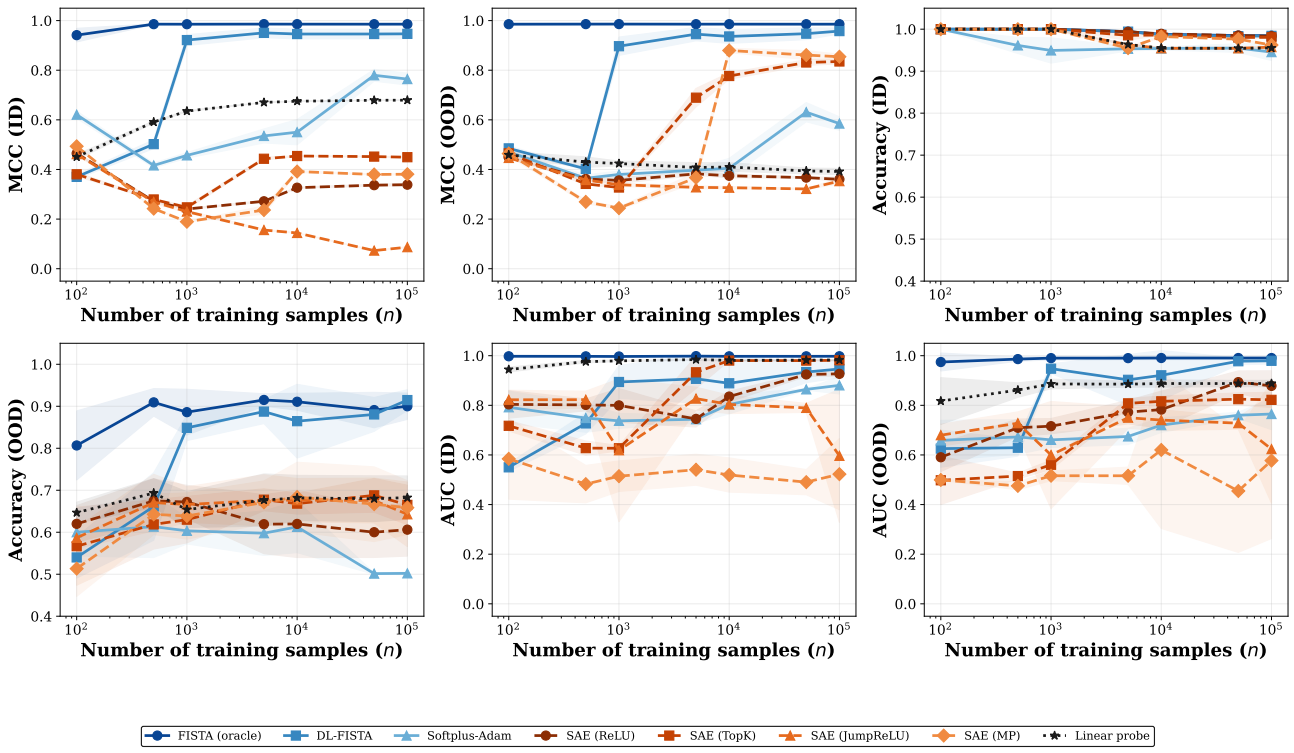


Figure 9: Performance vs number of training samples (n).

D.2 SCALING NUMBER OF SAMPLES

Figure 9.

D.3 DOES SPARSITY LEVEL OF THE LATENT FACTORS AFFECT THE GAP?

Figure 10

D.4 PHASE TRANSITION ABLATIONS

Figure 11

E THEORETICAL MODEL FOR TOY SETTING

We study the geometry of a system where a sparse source vector $z \in [0, 1]^3$ with at most two non-zero elements ($\|z\|_0 \leq 2$) is linearly projected to an observation $y \in \mathbb{R}^2$ (see Fig. **??**):

$$y = Az. \tag{11}$$

The sparsity constraint implies that any observation is a combination of at most two active source components. Whenever active, we assume that each source follows a uniform distribution $z_i \mid i$ active $\sim$ Uniform$(0, 1)$. The training data is considered *independent and identically distributed* (IID) and is generated from combinations of sources (z_1, z_2) or (z_2, z_3). The test data is considered *out-of-distribution* (OOD) and is generated from the novel combination (z_1, z_3). Our goal is to determine whether the first variable z_1 is above a certain, safety-relevant, threshold $z_1 = \frac{1}{2}$.

To analyze the geometry, we examine the columns $A_i \in \mathbb{R}^2$ of the projection matrix. We define the angles $\phi := \angle(A_1, A_2)$ and $\theta := \angle(A_1, A_3)$, which fully determine the system. To simplify the analysis, we make two assumptions:

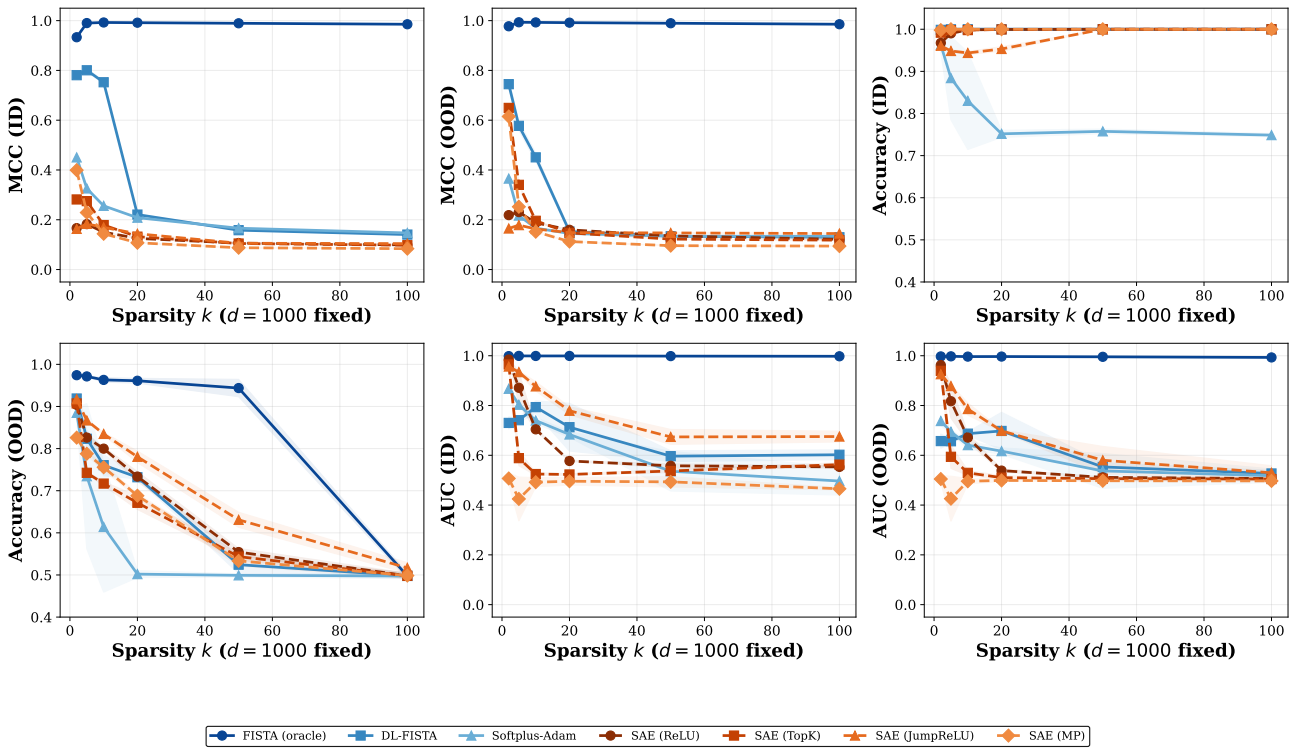


Figure 10: Performance vs sparsity (k).

1. We align our coordinate system and fix the magnitude of the first basis vector relative to our threshold, such that $A_1 = (2, 0)$ and $\|A_2\| = \|A_3\| = 1$.
2. To ensure the cones spanned by the vectors do not overlap, we require that $0 < \phi, \theta < \pi$ and $\phi + \theta > \pi$. This is an illustrative way of understanding why and when compressed sensing is possible in this system.

A perfect linear classifier trained on the IID data must separate the space based on the condition $z_1 = \frac{1}{2}$. In the observation space, this corresponds to a line parallel to A_2 and passing through the point $\frac{1}{2}A_1$. This decision boundary is the line parameterized by:

$$y(\beta) = \frac{1}{2}A_1 + \beta A_2, \quad \beta \in \mathbb{R}. \tag{12}$$

The question we are interested in is: *what is the accuracy of this classifier on the OOD data?* We derive the analytically predicted OOD accuracy for this perfect linear IID classifier, separating two cases, in Appendix E.1.

The simulations and analytical prediction are tested and illustrated in Fig. 16 confirming the validity of the theory.

E.1 DERIVATION

We study the geometry of a system where a sparse source vector $z \in [0, 1]^3$ with at most two non-zero elements ($\|z\|_0 \leq 2$) is linearly projected to an observation $y \in \mathbb{R}^2$ (see Fig. 1):

$$y = Az. \tag{13}$$

The sparsity constraint implies that any observation is a combination of at most two active source components. Whenever active, we assume that each source follows a uniform distribution $z_i \mid i$ active $\sim$ Uniform$(0, 1)$. The training data is considered *independent and identically distributed* (IID) and is generated from combinations of sources (z_1, z_2) or (z_2, z_3). The test data is considered *out-of-distribution* (OOD) and is generated from the novel combination (z_1, z_3). Our goal is to determine whether the first variable z_1 is above a certain, safety-relevant, threshold $z_1 = \frac{1}{2}$.

To analyze the geometry, we examine the columns $A_i \in \mathbb{R}^2$ of the projection matrix. We define the angles $\phi := \angle(A_1, A_2)$ and $\theta := \angle(A_1, A_3)$, which fully determine the system. To simplify the analysis, we make two assumptions:

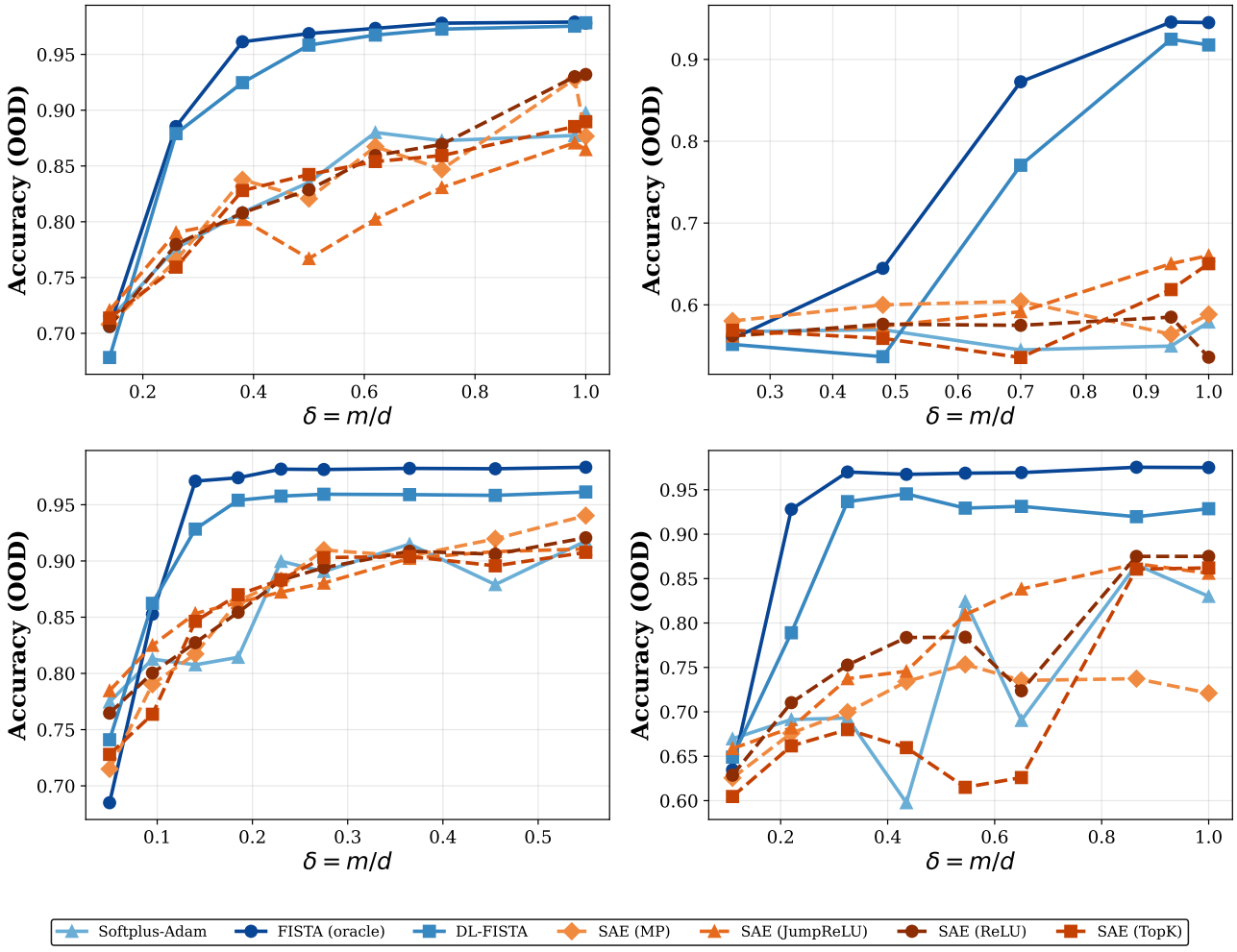


Figure 11: Phase transition: Accuracy (OOD) vs ρ.

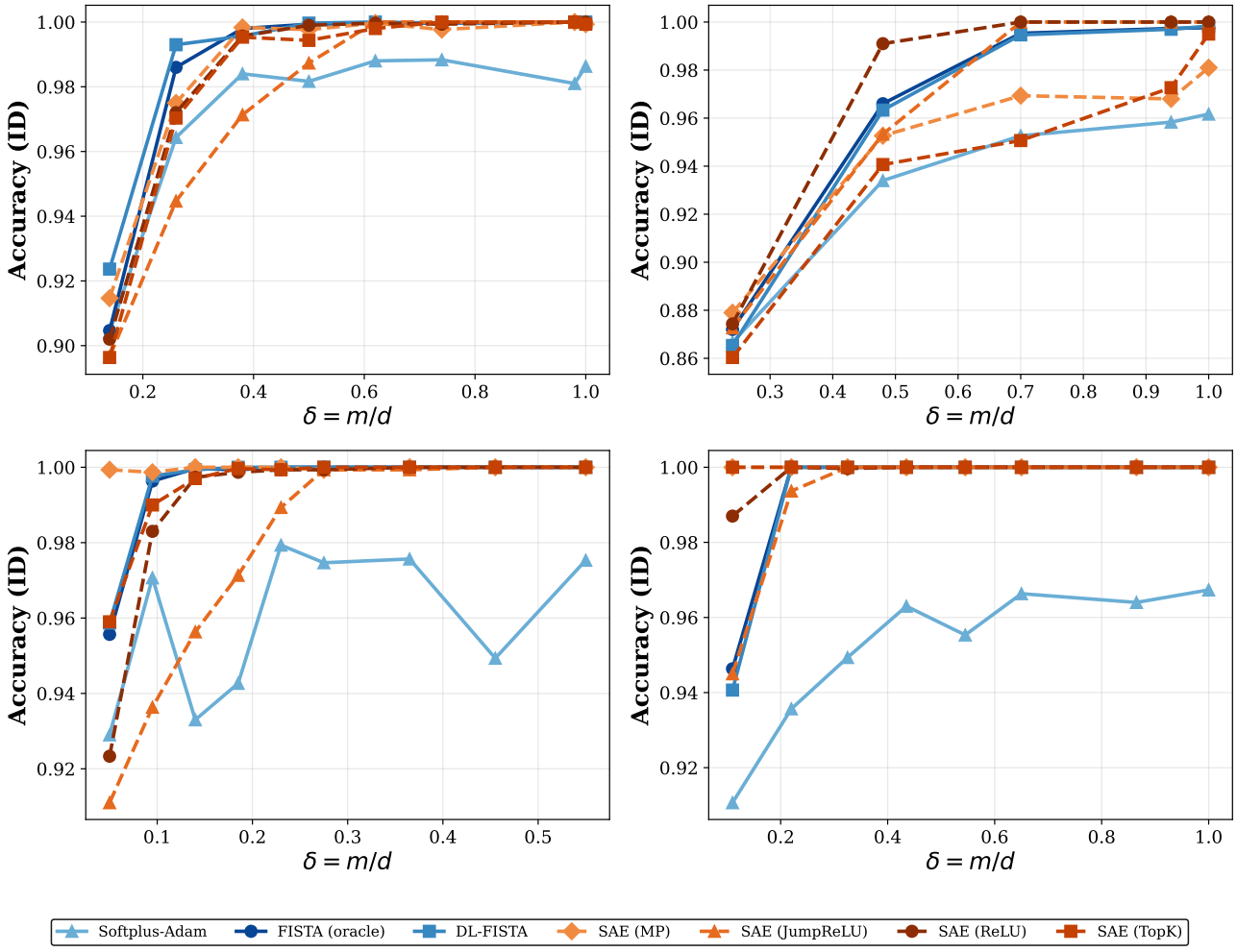


Figure 12: Phase transition: Accuracy (ID) vs ρ.

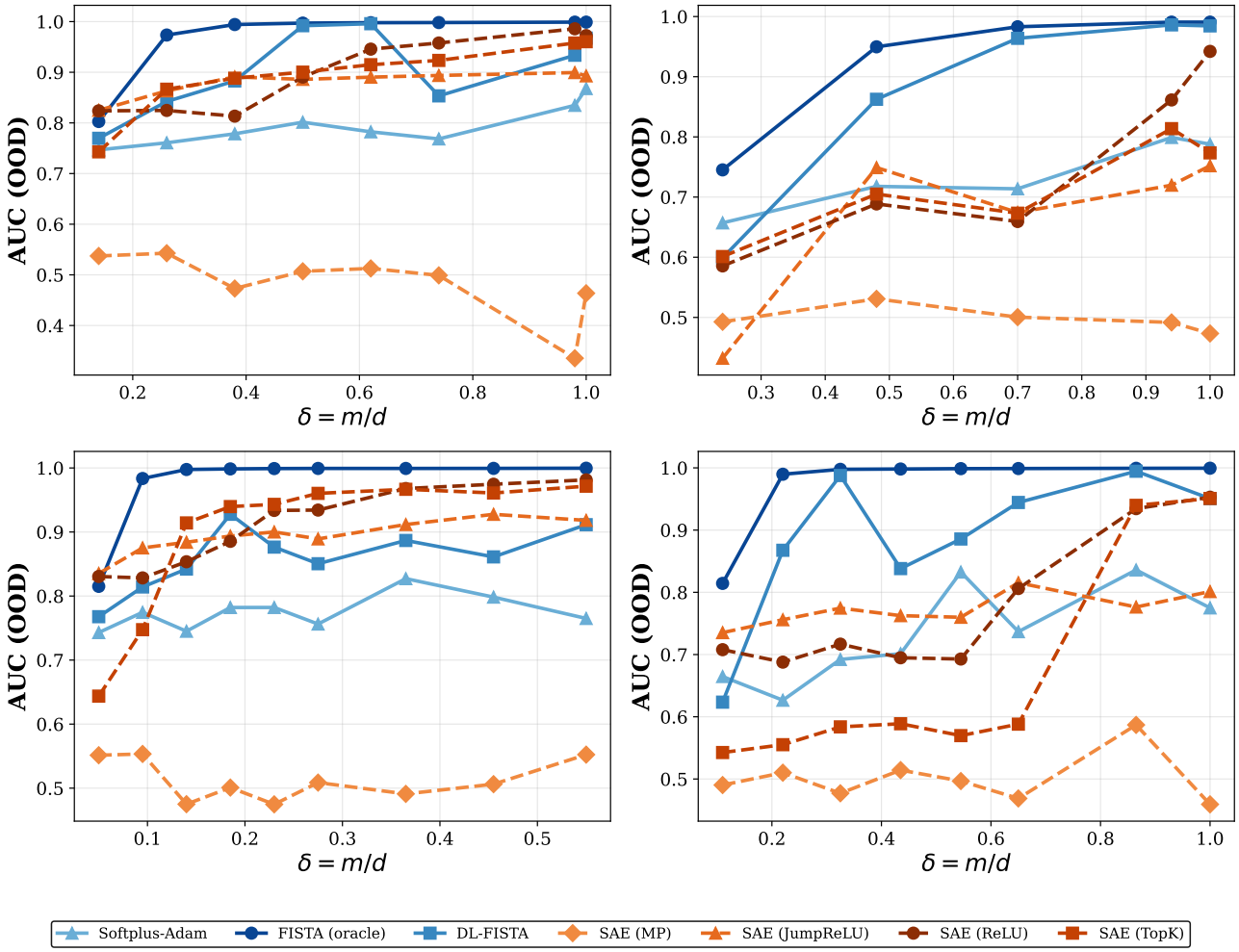


Figure 13: Phase transition: AUC (OOD) vs ρ.

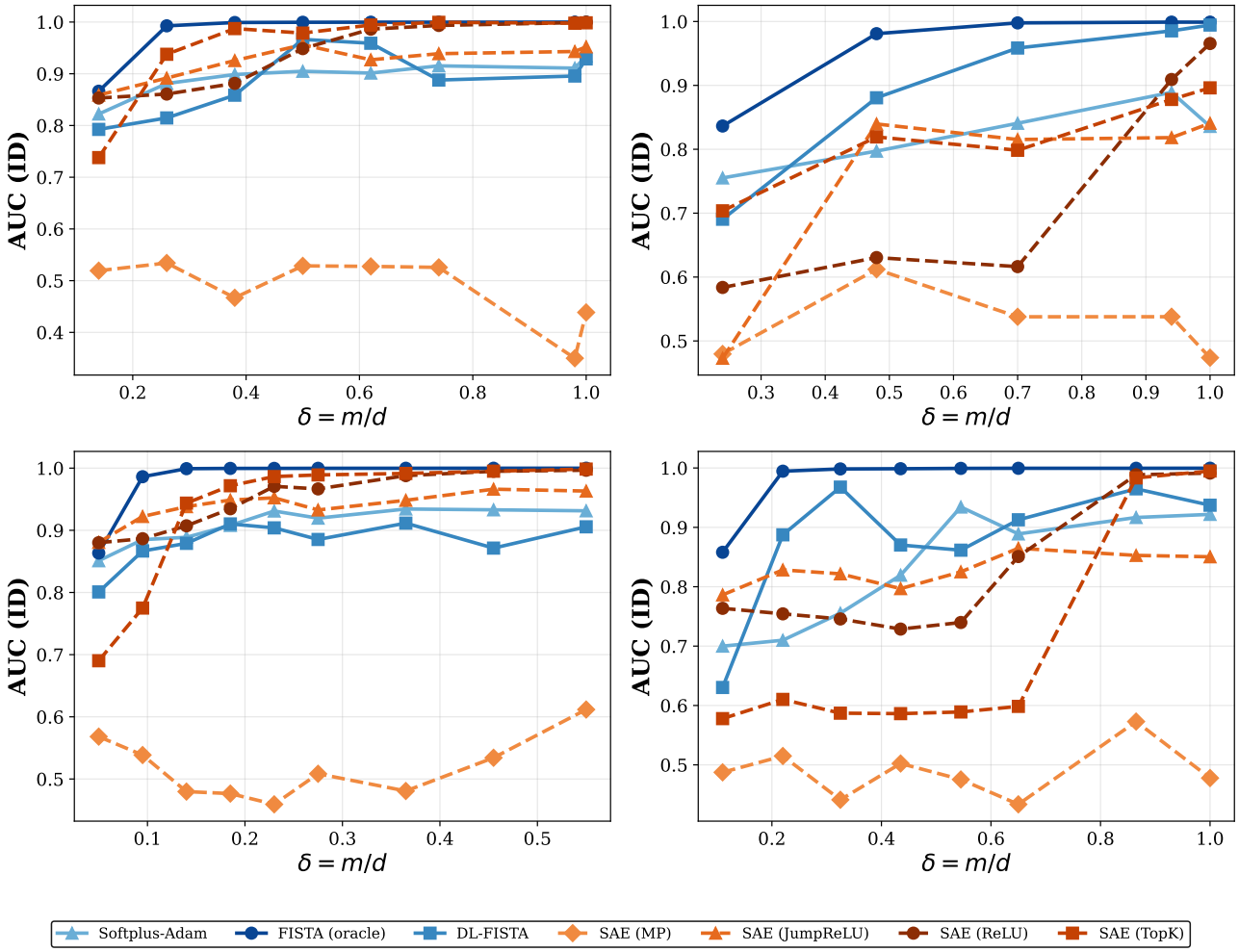


Figure 14: Phase transition: AUC (ID) vs ρ.

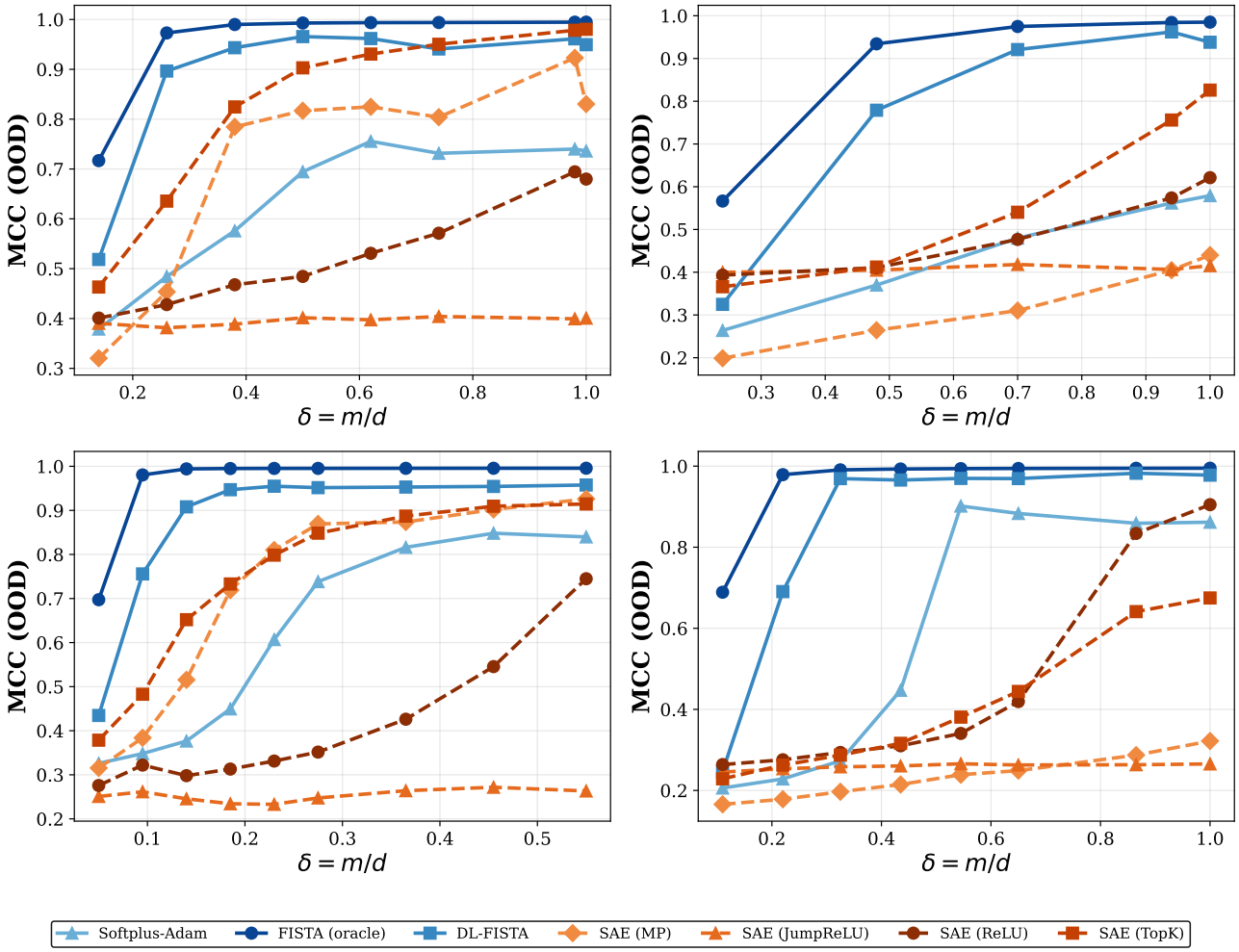


Figure 15: Phase transition: MCC (OOD) vs ρ.

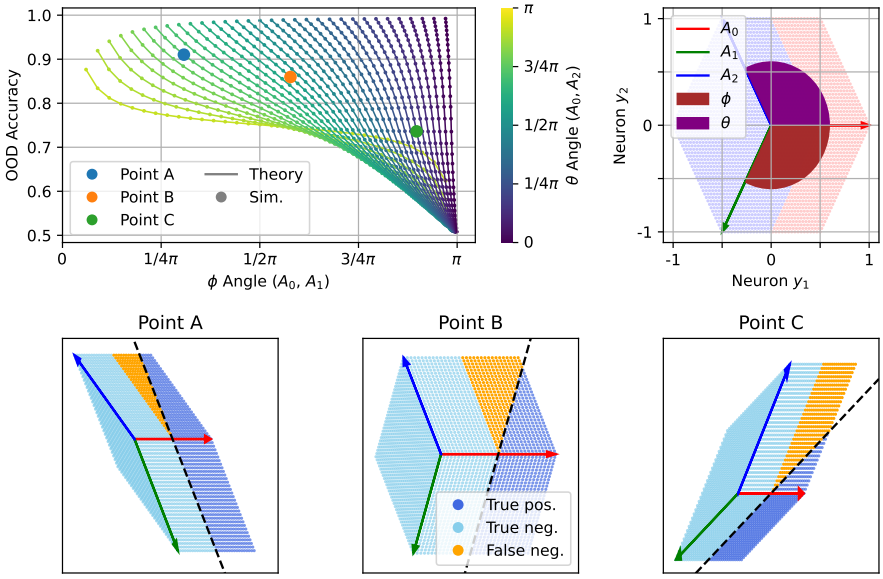


Figure 16: **Theory. Top left,** shows the theoretically predicted accuracy and simulations of a perfect linear classifier, trained and tested (OOD) on distinct latent combinations (see Fig. **??**). **Top right,** illustrates the geometry of the classification problem (red and blue classes) with the directions of the decoder A columns for each latent and the angles (ϕ, θ) between them. **Bottom,** shows the resulting geometry for three sample points from the first plot.

1. We align our coordinate system and fix the magnitude of the first basis vector relative to our threshold, such that $A_1 = (2, 0)$ and $\|A_2\| = \|A_3\| = 1$.
2. To ensure the cones spanned by the vectors do not overlap, we require that $0 < \phi, \theta < \pi$ and $\phi + \theta > \pi$. This is an illustrative way of understanding why and when compressed sensing is possible in this system.

A perfect linear classifier trained on the IID data must separate the space based on the condition $z_1 = \frac{1}{2}$. In the observation space, this corresponds to a line parallel to A_2 and passing through the point $\frac{1}{2}A_1$. This decision boundary is the line parameterized by:

$$y(\beta) = \frac{1}{2}A_1 + \beta A_2, \quad \beta \in \mathbb{R}. \tag{14}$$

The question we are interested in is: *what is the accuracy of this classifier on the OOD data?* Clearly, the perfect linear classifier for the OOD data would have a decision boundary that is parallel to A_3 and shifted by $\frac{1}{2}A_1$, i.e., the line:

$$y_{\text{OOD}}(\beta) = \frac{1}{2}A_1 + \beta A_3, \quad \beta \in \mathbb{R}. \tag{15}$$

Since $\phi + \theta > \pi$, we know that the IID classifier's boundary cannot be aligned with the ideal OOD classifier's boundary, so there must be some OOD error. Moreover, we know that the IID classifier can never 'under-shoot' on the OOD data (that would require $\phi + \theta < \pi$). Consequently, we will only observe *false negatives*—that is, test points with $z_1 > \frac{1}{2}$ that are erroneously classified as safe.

We now have to distinguish: **Case 1**, where the classifier passes right from the top right corner $(A_1 + A_3)$ (Fig. 16 Point C), and **Case 2**, where the classifier passes left from the top right corner $(A_1 + A_3)$ (Fig. 16 Point A).

The separation happens when the classifier passes through the top right corner. In that case it will form a triangle through the points $(\frac{1}{2}A_1, A_1, A_1 + A_3)$, with associated angles $(a, b, c) := (\pi - \phi, \pi - \theta, \phi + \theta - \pi)$. By assumption, the base of this triangle has length 1. Consequently, trigonometry tells us that the first angle must have a fixed relation to the second angle $a = \frac{\pi - b}{2}$. From this it follows that the condition for Case 1 is

$$\frac{\pi - (\pi - \theta)}{2} < \pi - \phi \quad \Rightarrow \quad \phi + \frac{\theta}{2} < \pi. \tag{16}$$

The total area of the right parallelogram $(\frac{1}{2}A_1, A_1, A_1 + A_3, \frac{1}{2}A_1 + A_3)$ is $\alpha = \sin(\theta)$. To compute the area of a triangle within this diagram, we use the fact that the area of a triangle can be computed from one side and the adjacent angles. We

will always pick a side with length 1, so that if the adjacent angles are (a, b), the area equals

$$\alpha(a, b) = \frac{\sin(a)\sin(b)}{2\sin(a+b)}. \tag{17}$$

In Case 1, we compute the area (α_1) of the triangle between the classifier and A_1. The base between $\frac{1}{2}A_1$ and A_1 has length 1. The angle on the left is $a_1 = \pi - \phi$ and the angle on the right is $b_1 = \pi - \theta$. Thus, using equation 17, the area is

$$\alpha_1 = \frac{\sin(\pi-\phi)\sin(\pi-\theta)}{2\sin(\phi+\theta-\pi)} = \frac{\sin(\phi)\sin(\theta)}{2\sin(\phi+\theta-\pi)} \tag{18}$$

The OOD accuracy will be 50% for the true negatives, plus 50% times the proportion that the area occupies in the right parallelogram (α)

$$acc_1(\text{OOD}) = \frac{1}{2} + \frac{\alpha_1}{2\alpha}. \tag{19}$$

In Case 2, we compute the area (α_2) of the triangle between the classifier and the correct OOD decision boundary. The base between $\frac{1}{2}A_1$ and $\frac{1}{2}A_1 + A_3$ has length 1. The angle on top is $a_2 = \pi - \theta$ and the angle below is $b_2 = \phi + \theta - \pi$. Thus, using equation 17, the area is

$$\alpha_2 = \frac{\sin(\pi-\theta)\sin(\phi+\theta-\pi)}{2\sin(\phi)} = \frac{\sin(\theta)\sin(\phi+\theta-\pi)}{2\sin(\phi)} \tag{20}$$

The OOD accuracy will be 100% minus the proportion that the area occupies in the left plus right parallelogram (2α)

$$acc_2(\text{OOD}) = 1 - \frac{\alpha_2}{2\alpha}. \tag{21}$$