A Unified Theory of Sparse Dictionary Learning in Mechanistic Interpretability: Piecewise Biconvexity and Spurious Minima

Anonymous authors
Paper under double-blind review

Abstract

As AI models achieve remarkable capabilities across diverse domains, understanding what representations they learn and how they encode concepts has become increasingly important for both scientific progress and trustworthy deployment. Recent works in mechanistic interpretability have widely reported that neural networks represent meaningful concepts as linear directions in their representation spaces and often encode diverse concepts in superposition. Various sparse dictionary learning (SDL) methods, including sparse autoencoders, transcoders, and crosscoders, are utilized to address this by training auxiliary models with sparsity constraints to disentangle these superposed concepts into monosemantic features. These methods are the backbone of modern mechanistic interpretability, yet in practice they consistently produce polysemantic features, feature absorption, and dead neurons, with very limited theoretical understanding of why these phenomena occur. Existing theoretical work is limited to tied-weight sparse autoencoders, leaving the broader family of SDL methods without formal grounding. We develop the first unified theoretical framework that casts all major SDL variants as a single piecewise biconvex optimization problem, and characterize its global solution set, non-identifiability, and spurious optima. This analysis yields principled explanations for feature absorption and dead neurons. To expose these pathologies under full ground-truth access, we introduce the Linear Representation Bench. Guided by our theory, we propose feature anchoring, a novel technique that restores SDL identifiability, substantially improving feature recovery across synthetic benchmarks and real neural representations, with direct implications for real-world applications on feature interpretation.

1 Introduction

As artificial intelligence systems scale to frontier capabilities, understanding their internal mechanisms has become essential for safe deployment (Lipton, 2017; Rudin, 2019). A central insight from mechanistic interpretability is that neural networks encode interpretable concepts as linear directions in superposition (Park et al., 2024; Elhage et al., 2022), where individual neurons respond to multiple unrelated concepts—a phenomenon known as polysemanticity (Bricken et al., 2023; Templeton et al., 2024). To disentangle these superposed representations, Sparse Dictionary Learning methods, including sparse autoencoders (SAEs) (Cunningham et al., 2023), transcoders (Dunefsky et al., 2024), and crosscoders (Lindsey et al., 2024), have emerged as the dominant paradigm, achieving remarkable empirical success on frontier language models (Templeton et al., 2024; Gao et al., 2024) and enabling applications from feature steering (Wang et al., 2025) to circuit analysis (Marks et al., 2025) and medical diagnosis (Abdulaal et al., 2024).

Despite this empirical success across diverse applications, SDL methods consistently exhibit persistent failure modes: learned features remain polysemantic (Chanin et al., 2025), "dead neurons" fail to activate on any data samples (Bricken et al., 2023), and "feature absorption" occurs where one neuron captures specific subconcepts while another responds to the remaining related concepts (Chanin et al., 2025). Practitioners have developed techniques to address these issues, including neuron resampling (Bricken et al., 2023), auxiliary losses (Gao et al., 2024), and careful hyperparameter tuning, yet these fixes remain ad-hoc engineering

solutions without principled justification. Critically, these phenomena persist even with careful training on clean data, suggesting they are not mere implementation artifacts but reflect fundamental structural properties of the SDL optimization problem itself, warranting rigorous theoretical analysis on root causes.

We argue that the root cause of these failure modes is the non-identifiability of SDL methods: even under the idealized Linear Representation Hypothesis, SDL optimization admits multiple solutions achieving perfect reconstruction loss, some recovering no interpretable ground-truth features at all, necessitating comprehensive theoretical analysis on SDL methods. While classical dictionary learning theory provides identifiability guarantees under strict conditions (Spielman et al., 2012; Gribonval & Schnass, 2010), and recent work establishes necessary conditions for tied-weight SAEs (Cui et al., 2025), no unified theoretical framework explains why diverse SDL methods, SAEs, transcoders, crosscoders, and their variants, systematically fail in predictable ways. Without theoretical grounding, the development of improved SDL methods remains largely empirical, potentially leaving highly effective techniques underexplored.

In this work, we develop a unified theoretical framework that formalizes SDL as a general optimization problem, encompassing various SDL methods (Bussmann et al., 2024; 2025; Tang et al., 2025b; Luo et al., 2023) as special cases. We demonstrate how these diverse methods instantiate our framework through different choices of input-output representation pairs, activation functions, and loss designs. We establish rigorous conditions under which SDL methods provably recover ground-truth interpretable features, characterizing the roles of feature sparsity, latent dimensionality, and activation functions. Through detailed analysis of the optimization landscape, we demonstrate that global minima correspond to correct feature recovery and provide necessary and sufficient conditions for achieving zero loss. We establish the prevalence of spurious partial minima exhibiting un-disentangled polysemanticity, providing novel theoretical explanations for feature absorption (Chanin et al., 2025) and the effectiveness of neuron resampling (Bricken et al., 2023). We design the Linear Representation Bench, a synthetic benchmark that strictly follows the Linear Representation Hypothesis, and evaluate diverse SDL methods with fully accessible ground-truth features, finding that Top-k family SAEs recover 84–90% of ground-truth features while ReLU-based methods consistently fail. Motivated by our theoretical insights, we propose feature anchoring, a technique applicable to all SDL methods that constrains learned features to known anchor directions, improving ground-truth recovery by up to 4.6 percentage points on synthetic benchmarks and lifting feature alignment to MIP > 0.85 on real CLIP representations, paving the way for more reliable utilization of SDL methods in mechanistic interpretability.

Our main contributions are as follows:

- We build the first theoretical framework for SDL in mechanistic interpretability as a general optimization problem encompassing diverse SDL methods.
- We theoretically prove that SDL optimization is biconvex, bridging mechanistic interpretability methods with traditional biconvex optimization theory.
- We characterize SDL optimization landscape and prove its non-identifiability, providing novel explanations for various phenomena observed empirically.
- We design the Linear Representation Bench, a benchmark with fully accessible ground-truth features, enabling fully transparent evaluation of SDL methods.
- We propose a novel technique, feature anchoring, that can achieve largely improved feature recovery performance applicable for all SDL methods. We validate the effectiveness of feature anchoring with extensive experiments across diverse SDL methods and settings.

2 Preliminaries

In this section, we present a unified theoretical framework for Sparse Dictionary Learning (SDL). We begin with the formal definitions of foundational concepts and our assumptions on representations. Then we introduce our framework and how various SDL methods instantiate it.

2.1 Input Distribution and Model Representation

Definition 2.1 (Input Distribution)**.** *Let $\mathcal{D}$ denote the distribution over possible inputs, $\mathcal{X}$, to a neural network. For example, $\mathcal{D}$ could be the distribution of natural images or the distribution of text sequences.*

Definition 2.2 (Model Representation)**.** *For a given model representation $\mathbf{x}$, let $n \in \mathbb{N}$ be its dimensionality. For each $s \sim \mathcal{D}$, the network produces a representation vector $\mathbf{x}(s)$, which is directly observable by running the model on s (We provide a full notation list in Appendix B).*

2.2 Assumptions

Empirical studies in mechanistic interpretability have observed that neural network representations encode meaningful concepts as linear directions, often in superposition (Marks & Tegmark, 2024; Nanda et al., 2023; Jiang et al., 2024; Park et al., 2025). Following Elhage et al. (2022) and Park et al. (2024), we primarily consider model representations $\mathbf{x}_p \in \mathbb{R}^{n_p}$ satisfying the following assumptions.

Assumption 2.1 (Representation Assumptions)**.** *There exists a* feature function $\mathbf{x} : \mathcal{X} \to \mathbb{R}^n$ *and a* feature matrix $W_p \in \mathbb{R}^{n_p \times n}$ *such that:*

1. ***Linear Decomposition**: For all $s \sim \mathcal{D}$,*

$$\mathbf{x}_p(s) = W_p \mathbf{x}(s).$$

2. ***Unit Norm**: The feature matrix $W_p \in \mathbb{R}^{n_p \times n}$ has unit-norm columns:*

$$\|W_p[:, i]\|_2 = 1 \quad \forall i \in [n]$$

3. ***Non-negativity**: $\mathbf{x}(s) \in \mathbb{R}^n_+$.*

4. ***Sparsity**: There exists $S \in [0, 1]$ such that $\forall i \in [n]$,*

$$\Pr_{s \sim \mathcal{D}}(x_i(s) = 0) \geq S.$$

We refer to $\mathbf{x}(s)$ as the ground-truth features*, where each $x_i(s)$ represents the activation level of concept i for input s. Researchers typically assume each component x_i of $\mathbf{x}$ corresponds to a human-interpretable concept (Bricken et al., 2023; Cunningham et al., 2023; Elhage et al., 2022).*

2.3 General Optimization Framework for SDL

Now we formalize SDL as an optimization problem under the Representation Assumptions (Assumption 2.1). We follow previous approaches Cui et al. (2025); Elhage et al. (2022) to omit the bias terms for mathematical simplicity.

Definition 2.3 (Sparse Dictionary Learning)**.** *A SDL model maps an input representation $\mathbf{x}_p(s) \in \mathbb{R}^{n_p}$ to a target representation $\mathbf{x}_r(s) \in \mathbb{R}^{n_r}$ through a two-layer architecture:*

(i) An encoder *layer that maps $\mathbf{x}_p(s)$ to a latent space:*

$$\mathbf{x}_q(s) = \sigma(W_E \mathbf{x}_p(s)) \tag{1}$$

(ii) A decoder *layer that maps the latents to $\mathbf{x}_r(s)$:*

$$\hat{\mathbf{x}}_r(s) = W_D \mathbf{x}_q(s) \tag{2}$$

where $\mathbf{x}_q(s) \in \mathbb{R}^{n_q}$, $W_E \in \mathbb{R}^{n_q \times n_p}$, $W_D \in \mathbb{R}^{n_r \times n_q}$, and $\sigma : \mathbb{R}^{n_q} \to \mathbb{R}^{n_q}$ is a sparsity-inducing activation function.

The SDL objective minimizes mean square error:

$$\mathcal{L}_{SDL} = \mathbb{E}_{s \sim \mathcal{D}} \left[\|\mathbf{x}_r(s) - W_D \sigma(W_E \mathbf{x}_p(s))\|_2^2 \right] \tag{3}$$

Under the Linear Representation Hypothesis (Assumption 2.1), both representations admit linear decompositions $\mathbf{x}_p(s) = W_p\mathbf{x}(s)$ and $\mathbf{x}_r(s) = W_r\mathbf{x}(s)$ in terms of ground-truth features $\mathbf{x}(s)$. The loss can thus be expressed as:

$$\mathcal{L}_{\text{SDL}} = \mathbb{E}_{s\sim\mathcal{D}}\left[\|W_r\mathbf{x}(s) - W_D\sigma(W_E W_p\mathbf{x}(s))\|_2^2\right] \tag{4}$$

2.4 Instantiations: Existing SDL Methods

We now demonstrate how existing SDL methods instantiate our framework through adopting different choices of input-target pairs $(\mathbf{x}_p, \mathbf{x}_r)$ and activation functions σ, and proposing variants on the loss function $\mathcal{L}_{\text{SDL}}$ (See Appendix A).

Sparse Autoencoders (SAEs). SAEs (Cunningham et al., 2023) decompose polysemantic activations into monosemantic components through sparsity constraints. In our framework, SAEs are characterized by setting $\mathbf{x}_r = \mathbf{x}_p$ (self-reconstruction). The encoder projects to a higher-dimensional sparse latent space, encouraging $\mathbf{x}_q(s)$ to capture the underlying ground-truth features $\mathbf{x}(s)$ (Figure 1).

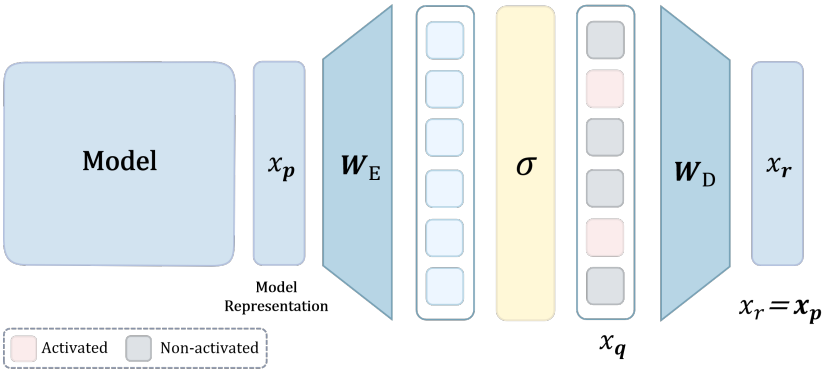

Figure 1: Sparse Autoencoder: encoder W_E maps $\mathbf{x}_p$ to sparse latents $\mathbf{x}_q$, decoder W_D reconstructs from $\mathbf{x}_q$.

Transcoders. Transcoders (Dunefsky et al., 2024; Paulo et al., 2025) capture interpretable features in layer-to-layer transformations. Unlike SAEs, transcoders approximate the input-output function of a target component, such as a MLP, using a sparse bottleneck. In our proposed theoretical framework, transcoders set $\mathbf{x}_p = \mathbf{x}_{\text{mid}}(s)$ and $\mathbf{x}_r = \mathbf{x}_{\text{pre}}(s)$, where $\mathbf{x}_{\text{mid}}(s)$ denotes the inputs of one MLP block, and $\mathbf{x}_{\text{pre}}(s)$ denotes the prediction of MLP's outputs (Figure 2).

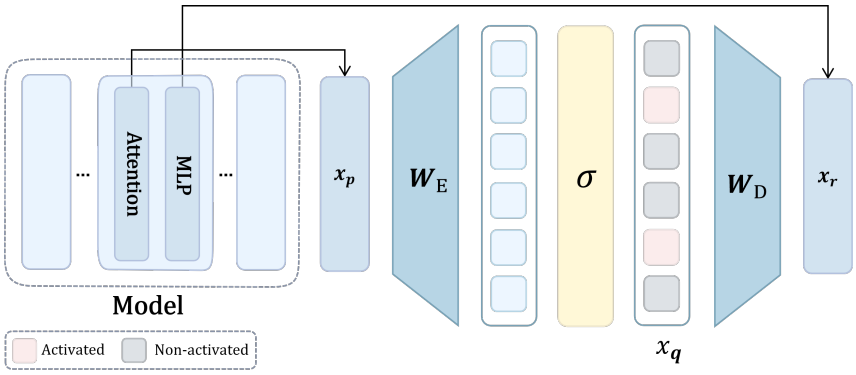

Figure 2: Transcoder: encoder W_E maps $\mathbf{x}_p(s)$ to sparse latents $\mathbf{x}_q(s)$, decoder W_D gives $\mathbf{x}_r(s)$ as a prediction of MLP's output.

Crosscoders. Crosscoders (Lindsey et al., 2024) discover shared features across multiple representation sources by jointly encoding and reconstructing concatenated representations. In our framework, crosscoders

set $\mathbf{x}_p = [\mathbf{x}_p^{(1)}; \ldots; \mathbf{x}_p^{(m)}]$ and $\mathbf{x}_r = [\mathbf{x}_r^{(1)}; \ldots; \mathbf{x}_r^{(m)}]$ where each superscript denotes a different source (Figure 3).

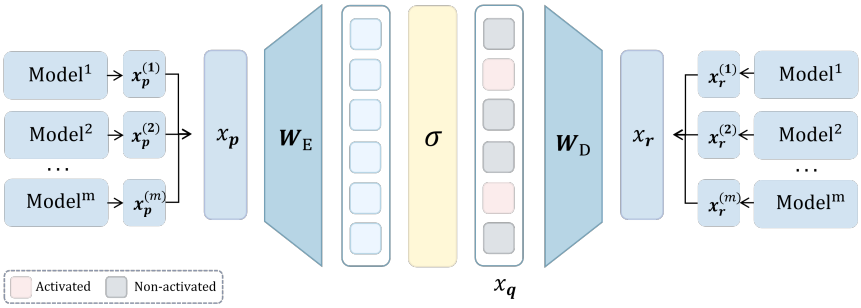

Figure 3: Crosscoder: encoder W_E maps concatenated multi-layer input $\mathbf{x}_p$ to $\mathbf{x}_q$, decoder W_D reconstructs multi-layer output $\mathbf{x}_r$.

Variants of SDL Methods. Various SDL methods fit into our theoretical framework but differ in their choices of activation functions and loss designs. Bricken et al. (2023) and Templeton et al. (2024) use ReLU activation $\sigma_{\text{ReLU}}(z) = \max(0, z)$ with L_1 regularization on latents:

$$\mathcal{L} = \mathbb{E}_{s\sim\mathcal{D}}\left[\|\mathbf{x}_r(s) - W_D\sigma_{\text{ReLU}}(W_E\mathbf{x}_p(s))\|_2^2 + \lambda\|\mathbf{x}_q(s)\|_1\right] \tag{5}$$

Rajamanoharan et al. (2024b) propose to utilize JumpReLU as the activation function:

$$\sigma_{\text{JumpReLU}}(z) = z \cdot H(z - \theta) \tag{6}$$

combined with a smoothed Heaviside function to directly penalize the L_0 norm. Makhzani & Frey (2014) introduce Top-k activation:

$$\sigma_{\text{Top-}k}(z)_i = \begin{cases} z_i & \text{if } z_i \text{ is among the } k \text{ largest components} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Bussmann et al. (2024) extend this to Batch Top-k activation, which applies Top-k selection across a batch to allow different samples to have different numbers of activated features. Rajamanoharan et al. (2024a) use ReLU activation with an additional Heaviside gating function. Gao et al. (2024) employ Top-k activation and introduce an auxiliary loss using Top-k_{aux} dead latents:

$$\begin{aligned}\mathcal{L} = \mathbb{E}_{s\sim\mathcal{D}}\Big[&\|\mathbf{x}_r(s) - W_D\sigma_{\text{Top-}k}(W_E\mathbf{x}_p(s))\|_2^2 \\ &+\lambda_{\text{aux}}\|\mathbf{x}_p(s) - W'_D\sigma_{\text{Top-}k_{\text{aux}}}(W_E\mathbf{x}_p(s))\|_2^2\Big]\end{aligned} \tag{8}$$

to resurrect dead neurons. Bussmann et al. (2025) and Tang et al. (2025b) use multiple k values with a multi-scale loss:

$$\mathcal{L} = \sum_{i=1}^{m} \lambda_i \mathbb{E}_{s\sim\mathcal{D}}\left[\|\mathbf{x}_r(s) - W_D\sigma_{\text{Top-}k_i}(W_E\mathbf{x}_p(s))\|_2^2\right] \tag{9}$$

that sums reconstruction errors of different sparsity levels.

The variants described above demonstrate that diverse SDL methods can be unified under our general framework (Definition 2.3) through specific choices of activation functions σ and loss modifications. Critically, this unification enables subsequent theoretical analysis to apply to the entire family of SDL methods, rather than just a single architecture. We also state a key property for SDL activation functions:

$$\sigma(z)_i \in \{0, z_i\} \quad \forall i \in [n_q] \tag{10}$$

This property holds for ReLU, JumpReLU, Top-k, Batch Top-k, and their compositions, the primary sparsity mechanisms used in practice.

3 Theoretical Results

Despite the empirical success of SDL methods across diverse applications, a significant gap exists between their practical use and our theoretical understanding of their optimization dynamics. This gap has important consequences: practitioners employ techniques like dead neuron resampling (Bricken et al., 2023) and observe phenomena like feature absorption (Chanin et al., 2025) without rigorous explanations for why these occur or how to systematically address them. Without theoretical grounding, the development of improved SDL methods remains largely empirical, potentially leaving highly effective techniques underexplored.

Our theoretical analysis bridges this gap by characterizing the SDL optimization landscape under the Linear Representation Hypothesis. Section 3.1 establishes a loss approximation enabling tractable analysis (Theorem 3.1). Section 3.2 proves SDL exhibits piecewise biconvex structure within activation pattern regions (Theorem 3.2). Section 3.3 characterizes the global minimum and shows the optimization is underdetermined (Theorem 3.3 and 3.4). Section 3.4 establishes the prevalence of spurious partial minima exhibiting polysemanticity (Theorem 3.6). Section 3.5 shows that hierarchical concept structures naturally induce feature absorption patterns that manifest as partial minima (Theorem 3.8). We provide full proofs in Appendix J.

3.1 Approximate Loss with Feature Reconstruction

We establish the bound for S satisfying $(1-S) \leq \frac{1}{n}$, the $\mathcal{L}_{\text{SDL}}$ can be approximated by per-feature reconstruction.

Theorem 3.1 (Loss Approximation)**.** *Under Assumption 2.1, let* $C = \sup_{s\sim\mathcal{D}} \|\mathbf{x}_r(s) - W_D\sigma(W_E\mathbf{x}_p(s))\|_2^2 < \infty$*. Define the approximate loss as:*

$$\tilde{\mathcal{L}}_{SDL}(W_D, W_E) := \sum_{d=1}^{n} M_d \left\| w_r^d - W_D\sigma(W_E w_p^d) \right\|^2 \tag{11}$$

where $M_d = \Pr(\mathbf{x}(s) = x_d(s)e_d) \cdot \mathbb{E}[x_d(s)^2 \mid x_d(s) > 0]$*. Then for any* S *satisfying* $(1-S) \leq \frac{1}{n}$*:*

$$\left|\mathcal{L}_{SDL} - \tilde{\mathcal{L}}_{SDL}\right| \leq n^2 C(1-S)^2 \tag{12}$$

This approximation isolates the contribution of each ground-truth feature, making the optimization landscape amenable to analysis techniques applied subsequently.

3.2 SDL Optimization is Piecewise Biconvex

While the SDL loss is non-convex globally due to activation discontinuities, it exhibits favorable convex structure within regions of fixed activation patterns.

Theorem 3.2 (Bi-convex Structure of SDL)**.** *Consider the approximate loss* $\tilde{\mathcal{L}}_{SDL}(W_D, W_E)$*. For an activation pattern* $\mathcal{P} = (\mathcal{F}_1, \ldots, \mathcal{F}_{n_q})$ *(See Definition 3.1), define the corresponding activation pattern region as:*

$$\begin{aligned}\Omega_{\mathcal{P}} = \Big\{ W_E \in \mathbb{R}^{n_q \times n_p} \;\Big|\; & \forall i \in [n_q], \\ & \mathcal{F}_i = \left\{ d \in [n] : \left(\sigma(W_E w_p^d)\right)_i > 0 \right\} \Big\},\end{aligned} \tag{13}$$

Then $\tilde{\mathcal{L}}_{SDL}$ *exhibits bi-convex structure over* $\mathbb{R}^{n_r \times n_q} \times \Omega_{\mathcal{P}}$*:*

1. *For any fixed* $W_E \in \Omega_{\mathcal{P}}$*,* $W_D \mapsto \tilde{\mathcal{L}}_{SDL}(W_D, W_E)$ *is convex in* W_D*.*
2. *For any fixed* $W_D \in \mathbb{R}^{n_r \times n_q}$*,* $W_E \mapsto \tilde{\mathcal{L}}_{SDL}(W_D, W_E)$ *is convex in* W_E *over* $\Omega_{\mathcal{P}}$*.*

This establishes SDL as a biconvex optimization problem within each activation pattern, bridging mechanistic interpretability with classical biconvex optimization theory, enabling both theoretical analysis of the optimization landscape and implementation of biconvex methods for SDL.

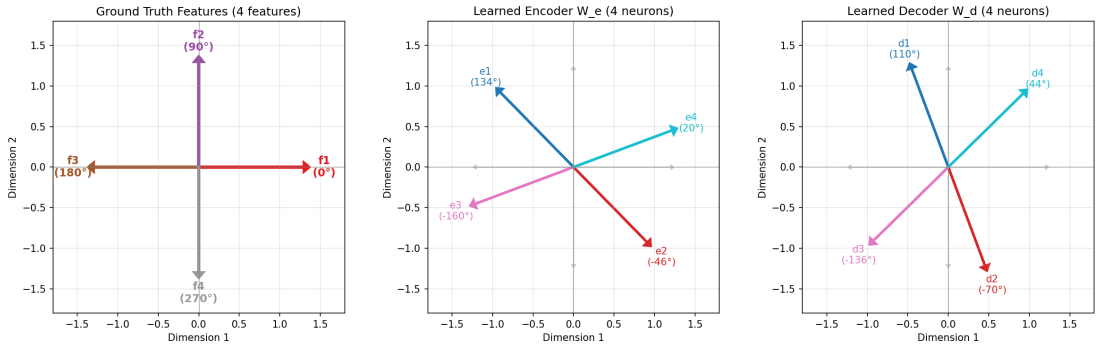

Figure 4: **Zero reconstruction loss without recovering ground-truth features.** We design the Linear Representation Bench that enable full knowledge of the ground truth features to study SDL methods. We observe one concerning phenomenon that these methods can achieve zero loss without recovering any ground truth features. **Left:** Four ground-truth feature directions. **Middle:** Learned encoder directions fail to align with ground truth. **Right:** Learned decoder directions are rotated accordingly. Although $\tilde{\mathcal{L}}_{\text{SDL}} \approx 0$, the learned features bear no correspondence to interpretable ground-truth concepts, demonstrating the underdetermined nature of SDL optimization.

3.3 Characterizing the Global Minimum of SDL

Characterizing the global minimum reveals both the success conditions for SDL training and the fundamental underdetermined nature of the optimization problem.

Theorem 3.3 (Successful Reconstruction Achieves Near-Zero Loss)**.** *Consider the approximate SDL loss*

$$\tilde{\mathcal{L}}_{SDL}(W_D, W_E) = \sum_{d=1}^{n} M_d \left\| w_r^d - W_D \sigma(W_E w_p^d) \right\|^2 \tag{14}$$

where $M_d = \Pr(\mathbf{x}(s) = x_d(s) e_d) \cdot \mathbb{E}[x_d(s)^2 \mid x_d(s) > 0]$*, and let* $M = \max_{i \neq j} \langle w_p^i, w_p^j \rangle$ *denote the maximum interference. When* $n_q \geq n$ *and* σ *satisfies* $\sigma(z)_i \in \{0, z_i\}$ *for all* i*, the configuration*

$$W_D^* = [W_r, \mathbf{0}], \quad W_E^* = \begin{bmatrix} W_p^\top \\ \mathbf{0} \end{bmatrix} \tag{15}$$

where $\mathbf{0}$ *denotes zero padding to dimension* n_q*, satisfies:*

$$\tilde{\mathcal{L}}_{SDL}(W_D^*, W_E^*) \leq n^2 M^2 \sum_{d=1}^{n} M_d \tag{16}$$

Theorem 3.4 (Necessary and Sufficient Conditions for Zero Loss)**.** *The approximate loss satisfies* $\tilde{\mathcal{L}}_{SDL}(W_D, W_E) = 0$ *if and only if*

$$w_r^d = W_D \sigma(W_E w_p^d) \quad \text{for all } d \in [n] \tag{17}$$

Theorem 3.3 provides a constructive global minimum that recovers ground-truth features, while Theorem 3.4 reveals the complete solution space. This system of n vector equations is underdetermined when $n_q > n$, admitting multiple solutions beyond the feature-recovering configuration. Critically, some solutions achieve zero reconstruction loss without recovering any ground-truth features (Figure 4).

3.4 Characterizing Spurious Partial Minima of SDL

Beyond the global minimum, SDL optimization exhibits spurious partial minima where neurons exhibit polysemanticity—responding to multiple unrelated features.

Example 3.5 (Spurious Partial Minimum)**.** *Consider $n = n_p = n_q = n_r = 2$, $\sigma = \sigma_{Top\text{-}1} \cdot \sigma_{ReLU}$, $S \to 1$, and $M_1 = M_2 = 1$. Let:*

$$w_p^1 = \begin{bmatrix}1\\0\end{bmatrix}, \quad w_p^2 = \begin{bmatrix}0\\1\end{bmatrix}, \quad w_r^1 = \begin{bmatrix}1\\0\end{bmatrix}, \quad w_r^2 = \begin{bmatrix}0\\1\end{bmatrix} \tag{18}$$

Consider the configuration where neuron 1 activates for both features while neuron 2 remains dead:

$$W_E^* = \begin{bmatrix}1 & 1\\0 & 0\end{bmatrix}, \quad W_D^* = \begin{bmatrix}1/2 & 0\\1/2 & 0\end{bmatrix} \tag{19}$$

Within the activation pattern region $\Omega_{\mathcal{A}} = \{W_E : \langle w_E^1, w_p^1\rangle > 0, \langle w_E^1, w_p^2\rangle > 0\}$, direct calculation shows $\nabla_{W_D}\tilde{\mathcal{L}} = 0$ and $\nabla_{W_E}\tilde{\mathcal{L}} = 0$. By traditional biconvex optimization theory (Gorski et al., 2007), (W_D^*, W_E^*) is a partial optimum. However, this configuration exhibits polysemanticity with suboptimal loss: $\tilde{\mathcal{L}}_{SDL}(W_D^*, W_E^*) = 1 > 0$.

Definition 3.1 (Activation Pattern)**.** *An* activation pattern *is a collection $\mathcal{P} = (\mathcal{F}_1, \ldots, \mathcal{F}_N)$ where $\mathcal{F}_i \subseteq [n]$ denotes the set of ground-truth features that activate neuron i.*

An activation pattern $\mathcal{P}$ is called:

- Polysemantic *if $\exists i \in [N]$ such that $|\mathcal{F}_i| \geq 2$ (at least one neuron responds to multiple features).*
- Realizable *if there exists an encoder $\sigma(W_E(\cdot)) \in \mathbb{R}^{N \times n_p}$ such that:*

$$\forall i \in [N], \quad \mathcal{F}_i = \left\{ d \in [n] : \left(\sigma(W_E w_p^d)\right)_i > 0 \right\} \tag{20}$$

That is, neuron i activates for exactly the features in $\mathcal{F}_i$ when they appear in isolation.

Theorem 3.6 (Prevalence of Spurious Partial Minima)**.** *Under Assumptions 2.1 with $n \geq 2$ and $n_q \geq n$, for any activation pattern $\mathcal{P} = (\mathcal{F}_1, \ldots, \mathcal{F}_{n_q})$ that is realizable, polysemantic, and forms a partition of $[n]$, there exists a partial minimum (W_D^*, W_E^*) of $\tilde{\mathcal{L}}_{SDL}$ exhibiting this pattern with positive loss.*

This establishes that partial minima are pervasive in SDL optimization: every realizable polysemantic activation pattern corresponds to a partial optima point where gradient descent can become trapped, a persistent challenge for SDL.

3.5 Theoretical Explanation for Feature Absorption

Feature absorption—where one neuron captures, or "absorbs", a specific sub-concept while another responds to remaining related concepts—frequently occurs in SDL training (Chanin et al., 2025). Though prevalently encountered and unwanted, researchers have limited understanding about why feature absorption occurs in SDL training. Here we show hierarchical concept structures naturally introduce realizable activation patterns exhibiting feature absorption and therefore connected with the framework's partial minima.

Example 3.7 (Feature Absorption)**.** *Consider a representation space with a parent concept "Dog" and four sub-concepts: "Border Collie", "Golden Retriever", "Husky", and "German Shepherd". Ideally, SDL learns separate monosemantic neurons for Dog, Cat, Horse, and Elephant, with each dog breed activating only the Dog neuron. However, feature absorption can result in the pattern in Figure 5: one neuron exclusively captures "Border Collie" (absorbed feature), while another responds to the remaining three breeds collectively (main line interpretation).*

Definition 3.2 (Hierarchical Concept Structure)**.** *A set of ground-truth features exhibits* hierarchical structure *if it is composed of a parent concept p and a set of sub-concepts $c_1, \ldots, c_k$ satisfying: $p(x) > 0 \iff \exists i \in [k], c_i(x) > 0$.*

Theorem 3.8 (Feature Absorption from Hierarchical Structure)**.** *Suppose there exist M parent concepts with hierarchical decompositions into sub-concepts: for each $i \in [M]$, parent concept d_i decomposes into sub-concepts $\mathcal{F}_i = \{d_{i,1}, \ldots, d_{i,k_i}\}$ where $k_i \geq 2$.*

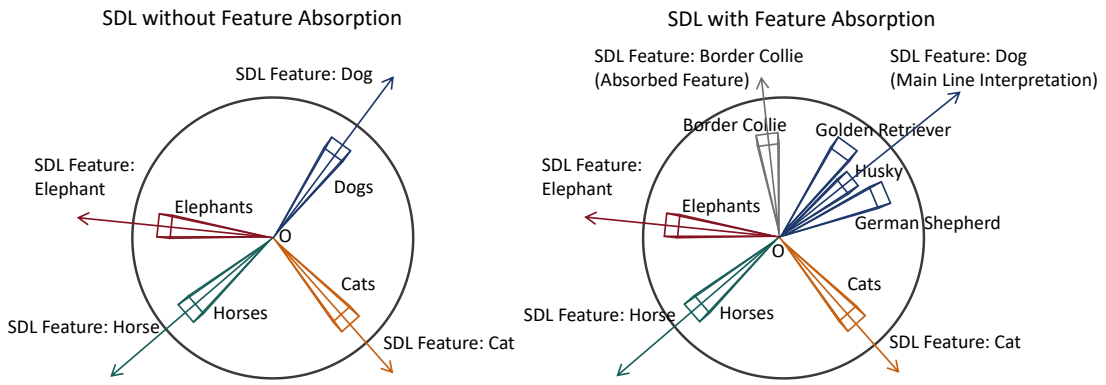

Figure 5: **Feature absorption emerges from hierarchical concept structure.** Left: Ideal SDL features without absorption. Right: hierarchical concept structure exists and only a proportion of the sub-concepts of "Dog" can activate the SDL feature.

If the activation pattern $\mathcal{P} = (\mathcal{F}_1, \ldots, \mathcal{F}_M)$ *is realizable, then* $\forall i^* \in [M]$, $\exists j^* \in [k_{i^*}]$ *such that the following activation pattern exhibiting feature absorption is realizable:*

$$\mathcal{P}' = (\mathcal{F}_1, \ldots, \mathcal{F}_{i^*} \setminus \{d_{i^*,j^*}\}, \{d_{i^*,j^*}\}, \ldots, \mathcal{F}_M) \tag{21}$$

Theorem 3.8 explains why feature absorption persists even with careful training: hierarchical concept structures naturally induce realizable polysemantic patterns that manifest as partial minima, providing theoretical grounding for this widely observed empirical phenomenon.

4 Method

Our theoretical analysis reveals a fundamental challenge in SDL optimization: the underdetermined nature of the loss landscape (Theorem 3.4), and the solution space admits multiple configurations—some achieving zero reconstruction loss without recovering any interpretable ground-truth features (Figure 4). These theoretical findings motivate our method, **feature anchoring**, a technique that constrains a subset of features to align with known semantic directions.

4.1 Anchor Feature Extraction

Feature anchoring requires identifying k anchor features $\{\tilde{\mathbf{w}}_p^{(i)}, \tilde{\mathbf{w}}_r^{(i)}\}_{i=1}^k$ that represent semantically meaningful directions in the representation space. We present two methods for obtaining these anchors:

Ground-Truth Features (Linear Representation Bench). When ground-truth features are available—as in our Linear Representation Bench where features are known by construction—we directly use a random subset of k ground-truth feature directions:

$$\tilde{w}_p^{(i)} = W_p^{\text{true}}[:, i], \quad \tilde{w}_r^{(i)} = W_r^{\text{true}}[:, i], \quad i \in \mathcal{K} \tag{22}$$

where $\mathcal{K} \subset [n]$ is a randomly selected subset of size k.

Subpopulation Mean Embeddings. For real-world datasets where ground-truth features are unknown, we identify semantic subpopulations Luo et al. (2024b) and compute their mean representations. Specifically, given a labeled dataset $\mathcal{D} = \{(s_j, y_j)\}_{j=1}^N$, where $y_j \in \{1, \ldots, C\}$, we compute the mean of representations for one subpopulation cluster, which is considered as a proxy for ground-truth features:

$$\bar{\mathbf{x}}_p^{(c)} = \frac{1}{|\{j : y_j = c\}|} \sum_{j: y_j = c} \mathbf{x}_p(s_j) \tag{23}$$

Then we normalize each mean representation to obtain anchor directions:

$$\tilde{\mathbf{w}}_p^{(c)} = \frac{\bar{\mathbf{x}}_p^{(c)}}{\|\bar{\mathbf{x}}_p^{(c)}\|_2} \tag{24}$$

4.2 SDL Loss Function with Feature Anchoring

Given k anchor features $\{\tilde{\mathbf{w}}_p^{(i)}, \tilde{\mathbf{w}}_r^{(i)}\}_{i=1}^k$, we modify the SDL optimization objective to include an anchoring penalty that constrains the first k encoder rows and decoder columns to align with these anchors.

The complete anchored SDL objective is:

$$\mathcal{L}_{\text{SDL-FA}} = \mathcal{L}_{\text{SDL}} + \lambda_{\text{anchor}} \mathcal{L}_{\text{anchor}} \tag{25}$$

where $\mathcal{L}_{\text{SDL}}$ is the standard SDL loss (Equation 3) and the anchoring loss is:

$$\begin{aligned} \mathcal{L}_{\text{anchor}} = &\|W_E[1{:}k, :] - [\tilde{w}_p^{(1)}, \ldots, \tilde{w}_p^{(k)}]^\top\|_F^2 \\ &+ \|W_D[:, 1{:}k] - [\tilde{w}_r^{(1)}, \ldots, \tilde{w}_r^{(k)}]\|_F^2 \end{aligned} \tag{26}$$

This feature anchoring technique (Equation 26) reduces the underdetermined nature of SDL training, and is *method-agnostic*: it applies equally to SAEs, transcoders, crosscoders, and their variants (TopK SAEs, Matryoshka SAEs, etc.) since it only constrains the encoder W_E and decoder W_D matrices that are common to all SDL architectures. We believe this universality makes feature anchoring a broadly useful technique for improving performance across SDL methods.

5 Experimental Results

We first evaluate feature anchoring with various SDL methods on the Linear Representation Bench (Section 5.1), a synthetic benchmark with fully accessible ground-truth features that precisely instantiates our theoretical assumptions. Second, we validate feature anchoring on real-world data by training SDL methods on CLIP embeddings of ImageNet-1K. Third, we provide empirical evidence showing that dead neuron resampling helps escape spurious local minima in large language models. Ablation studies are in Appendix H.

5.1 Results on The Linear Representation Bench

To validate our theoretical predictions, we design the Linear Representation Bench, a benchmark that precisely instantiates Assumptions 2.1 with fully known GT features.

Data Generation. We generate synthetic representations $\mathbf{x}_p(s) = W_p^{\text{true}}\mathbf{x}(s)$ where ground-truth features $\mathbf{x}(s) \in \mathbb{R}^n$ follow shifted exponential distributions with sparsity $\mathcal{S}$. The feature matrix $W_p^{\text{true}} \in \mathbb{R}^{n_p \times n}$ is constructed via gradient-based optimization (Detailed in Appendix C).

Metrics. Given learned features $W_p^{\text{learned}} = W_E^\top \in \mathbb{R}^{n_p \times n_q}$ and ground truth features $W_p^{\text{true}} \in \mathbb{R}^{n_p \times n}$ (columns are unit-norm feature directions), we compute the similarity matrix $\mathbf{S} = |W_p^{\text{learned}\top} W_p^{\text{true}}| \in \mathbb{R}^{n_q \times n}$. For each ground truth feature $i \in [n]$, we find its best match score: $s_i = \max_j S_{ji}$.

- **GT Recovery** is the fraction of ground truth features recovered ($s_i > \tau$), $\mathbf{M_{GT}}(W_p^{\text{learned}}) = \frac{1}{n}\sum_{i=1}^n \mathbf{1}\{s_i > \tau\}$. Anchor features are held out and only non-anchor features are used for evaluation.
- **Maximum Inner Product** is the mean of best match scores: $\mathbf{M_{IP}}(W_p^{learned}) = \frac{1}{n}\sum_{i=1}^n s_i$.

As shown in Table 1 and Figure 6, feature anchoring significantly improves feature recovery performances.

Table 1: Feature recovery results on the Linear Representation Bench ($n = 1000$, $n_p = n_r = 768$, $n_q = 16000$, $\mathcal{S} = 0.99$). Evaluation is performed only on features not used for anchoring. Feature anchoring (FA) consistently improves both GT Recovery ($\mathbf{M_{GT}}$) and Maximum Inner Product ($\mathbf{M_{IP}}$) across all SDL methods. *: For some methods, $\mathbf{M_{GT}}$ is 0% because all best-match similarities fall below the evaluation threshold (0.95). In these cases, $\mathbf{M_{IP}}$, which remains well-aligned with $\mathbf{M_{GT}}$, serves as a more informative indicator of recovery performance. See Figure 6 for a threshold-dependent view of $\mathbf{M_{GT}}$.

Method	$\mathbf{M_{GT}} \uparrow$	$\mathbf{M_{IP}} \uparrow$
ReLU SAE	0.00%*	0.205
+ Feature Anchoring	0.00%*	**0.246**
JumpReLU SAE	0.00%*	0.237
+ Feature Anchoring	0.00%*	**0.333**
TopK SAE	84.90%	0.983
+ Feature Anchoring	**87.63%**	**0.986**
BatchTopK SAE	84.80%	0.981
+ Feature Anchoring	**89.38%**	**0.988**
Matryoshka SAE	83.70%	0.982
+ Feature Anchoring	**87.32%**	**0.985**
Transcoder	23.60%	0.838
+ Feature Anchoring	**25.05%**	0.838
Crosscoder	56.42%	0.940
+ Feature Anchoring	**57.71%**	**0.941**

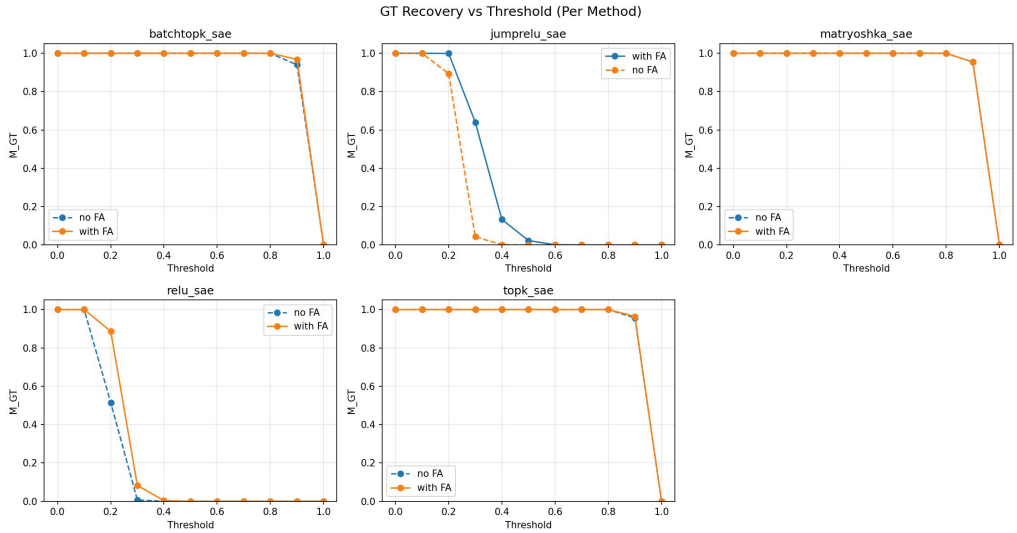

Figure 6: **$\mathbf{M_{GT}}$ vs. Threshold.** Across all SDL methods, feature anchoring consistently improves feature recovery, as measured by $\mathbf{M_{GT}}$ over a range of thresholds.

5.2 Results on CLIP Embeddings of ImageNet

To validate that feature anchoring generalizes beyond synthetic benchmarks to real-world representations, we apply our method to CLIP embeddings of ImageNet-1K (Russakovsky et al., 2015). We extract CLIP-ViT-B/32 (Radford et al., 2021) image embeddings for all ImageNet training images and compute ground-truth anchor features as normalized class mean embeddings: $\tilde{w}_p^{(c)} = \bar{x}_p^{(c)} / \|\bar{x}_p^{(c)}\|_2$ where $\bar{x}_p^{(c)}$ is the mean embedding over all images in class c. We train TopK, BatchTopK, and Matryoshka SAEs with $n_p = n_r = 768$,

$n_q = 16,384$, and use $k = 30$ anchors with $\lambda_{\text{anchor}} = 1.0$. Table 2 shows feature anchoring consistently improves feature recovery (Examples in Appendix I).

Table 2: Feature recovery on CLIP embeddings. As ground-truth features are not explicitly available in this setting, we use subpopulation mean embeddings as a proxy for the underlying feature directions. *: 0% recovery resulted from the same issue as Talble 1.

Method	$\mathbf{M_{GT}} \uparrow$	$\mathbf{M_{IP}} \uparrow$
TopK SAE	0.00%*	0.517
+ Feature Anchoring	**24.13%**	**0.851**
BatchTopK SAE	0.00%*	0.508
+ Feature Anchoring	**24.13%**	**0.847**
Matryoshka SAE	0.00%*	0.683
+ Feature Anchoring	**24.45%**	**0.858**

5.3 Neuron Resampling Helps Escape Partial Minima

Our partial minima analysis (Theorem 3.6) proves the prevalence of spurious partial minima and connects them to dead neurons ($\mathcal{F}_i = \emptyset$). To address this, we utilize neuron resampling (Bricken et al., 2023) to help SDL training overcome these partial minima. We argue that resampling reinitializes dead neurons toward under-reconstructed directions, perturbing the optimization away from the partial minima.

To validate this, we train SAEs on Llama 3.1 8B Instruct (layer 12, dimension 4096) with latent dimension 131,072 for 30,000 steps on FineWeb-Edu (Penedo et al., 2024), processing 4096 token activations per step. We compare standard training against feature resampling after 5000 steps. As shown in Figure 7, resampling enables the optimizer to escape spurious local minima, achieving lower final loss.

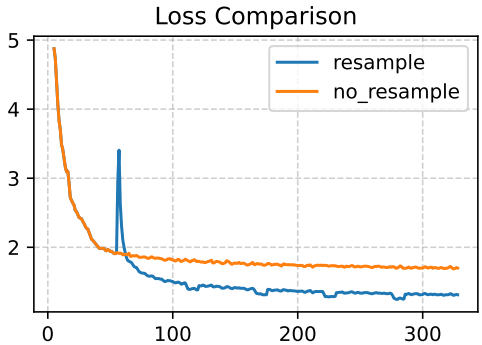

Figure 7: **Feature resampling accelerates convergence and improves final loss.** Training curves on Llama 3.1 8B comparing standard SAE training (blue) with periodic dead neuron resampling after 5000 steps (in the plot x axis is scaled by 100).

6 Conclusion

We develop the first unified theoretical framework for Sparse Dictionary Learning in mechanistic interpretability, demonstrating how diverse SDL methods instantiate a single optimization problem. We prove that SDL exhibits piecewise biconvex structure, bridging mechanistic interpretability with classical optimization theory. We characterize the global minimum and establish that the optimization is fundamentally underdetermined, admitting solutions that achieve zero reconstruction loss without recovering interpretable features. We demonstrate that spurious partial minima exhibiting polysemanticity are pervasive, and prove that hierarchical concept structures naturally induce feature absorption patterns that manifest as partial minima. To validate our theory, we design the Linear Representation Bench with fully accessible ground-truth features, and propose feature anchoring—a technique applicable to all SDL methods that addresses the underdetermined nature of optimization.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Code Availability

We will provide full scripts of the Linear Representation Bench and feature anchoring upon acceptance.

Declaration of LLM Usage

We admit the usage of Claude Sonnet-4.6 for coding and polishing the manuscript.

References

Ahmed Abdulaal, Hugo Fry, Nina Montaña-Brown, Ayodeji Ijishakin, Jack Gao, Stephanie Hyland, Daniel C. Alexander, and Daniel C. Castro. An x-ray is worth 15 features: Sparse autoencoders for interpretable radiology report generation, 2024. URL https://arxiv.org/abs/2410.03334.

Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006.

Leonard Bereska and Efstratios Gavves. Mechanistic interpretability for ai safety – a review, 2024. URL https://arxiv.org/abs/2404.14082.

Steven Bills, Nick Cammarata, Dan Danworth, Chris Olah, Brandon Witt, and Jeff Wu. Language models can explain neurons in language models. *OpenAI Blog*, 2023.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nicholas L Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Chris Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features.

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

Bart Bussmann, Noa Nabeshima, Adam Karvonen, and Neel Nanda. Learning multi-level features with matryoshka sparse autoencoders, 2025. URL https://arxiv.org/abs/2503.17547.

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

Hakaze Cho, Haolin Yang, Brian M. Kurkoski, and Naoya Inoue. Binary autoencoder for mechanistic interpretability of large language models, 2025. URL https://arxiv.org/abs/2509.20997.

Jingyi Cui, Qi Zhang, Yifei Wang, and Yisen Wang. On the theoretical understanding of identifiable sparse autoencoders and beyond, 2025. URL https://arxiv.org/abs/2506.15963.

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.

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

Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. Transcoders find interpretable llm feature circuits, 2024. URL https://arxiv.org/abs/2406.11944.

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

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

R. Farrell et al. Editing and unlearning with sparse autoencoders. *arXiv preprint arXiv:2410.19278*, 2024.

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.

Rong Ge, Jason D. Lee, and Tengyu Ma. Matrix completion has no spurious local minimum, 2018. URL https://arxiv.org/abs/1605.07272.

Jochen Gorski, Frank Pfeuffer, and Kathrin Klamroth. Biconvex sets and optimization with biconvex functions: a survey and extensions. *Mathematical methods of operations research*, 66(3):373–407, 2007.

Remi Gribonval and Karin Schnass. Dictionary identification - sparse matrix-factorisation via ℓ_1-minimisation, 2010. URL https://arxiv.org/abs/0904.4774.

Onkar Gujral, Mihir Bafna, Eric Alm, and Bonnie Berger. Sparse autoencoders uncover biologically interpretable features in protein language model representations. *Proceedings of the National Academy of Sciences*, 122(34):e2506316122, 2025.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Krasheninnikov, and Jacob Steinhardt. Finding neurons in a haystack: Case studies in phenomenological interpretability. *arXiv preprint arXiv:2305.01610*, 2023.

Prateek Jain and Purushottam Kar. Non-convex optimization for machine learning. *Foundations and Trends® in Machine Learning*, 10(3–4):142–336, 2017. ISSN 1935-8245. doi: 10.1561/2200000058. URL http://dx.doi.org/10.1561/2200000058.

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.

Adam Karvonen, Benjamin Wright, Can Rager, Rico Angell, Jannik Brinkmann, Logan Smith, Claudio Mayrink Verdun, David Bau, and Samuel Marks. Measuring progress in dictionary learning for language model interpretability with board game models, 2024. URL https://arxiv.org/abs/2408.00113.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav), 2018. URL https://arxiv.org/abs/1711.11279.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models, 2020. URL https://arxiv.org/abs/2007.04612.

Daniel D Lee, P Pham, Y Largman, and A Ng. Advances in neural information processing systems 22. *Tech Rep*, 2009.

Jack Lindsey, Adly Templeton, Jonathan Marcus, Thomas Conerly, Joshua Batson, and Christopher Olah. Sparse crosscoders for cross-layer features and model diffing. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/crosscoders/index.html.

Zachary C. Lipton. The mythos of model interpretability, 2017. URL https://arxiv.org/abs/1606.03490.

Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions, 2017. URL https://arxiv.org/abs/1705.07874.

Yifan Luo, Yiming Tang, Chengfeng Shen, Zhennan Zhou, and Bin Dong. Prompt engineering through the lens of optimal control, 2023. URL https://arxiv.org/abs/2310.14201.

Yulin Luo, Ruichuan An, Bocheng Zou, Yiming Tang, Jiaming Liu, and Shanghang Zhang. Llm as dataset analyst: Subpopulation structure discovery with large language model.

Yulin Luo, Ruichuan An, Bocheng Zou, Yiming Tang, Jiaming Liu, and Shanghang Zhang. Llm as dataset analyst: Subpopulation structure discovery with large language model. In *European Conference on Computer Vision*, pp. 235–252. Springer, 2024a.

Yulin Luo, Ruichuan An, Bocheng Zou, Yiming Tang, Jiaming Liu, and Shanghang Zhang. Llm as dataset analyst: Subpopulation structure discovery with large language model, 2024b. URL https://arxiv.org/abs/2405.02363.

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*, pp. 689–696, 2009.

Alireza Makhzani and Brendan Frey. k-sparse autoencoders, 2014. URL https://arxiv.org/abs/1312.5663.

Ziming Mao, Jia Xu, Zeqi Zheng, Haofang Zheng, Dabing Sheng, Yaochu Jin, and Guoyuan Yang. Sparse autoencoders bridge the deep learning model and the brain, 2025. URL https://arxiv.org/abs/2506.11123.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets, 2024. URL https://arxiv.org/abs/2310.06824.

Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models, 2025. URL https://arxiv.org/abs/2403.19647.

Ibomoiye Domor Mienye and Nobert Jere. A survey of decision trees: Concepts, algorithms, and applications. *IEEE access*, 12:86716–86727, 2024.

Aaron Mueller, Atticus Geiger, Sarah Wiegreffe, Dana Arad, Iván Arcuschin, Adam Belfki, Yik Siu Chan, Jaden Fiotto-Kaufman, Tal Haklay, Michael Hanna, Jing Huang, Rohan Gupta, Yaniv Nikankin, Hadas Orgad, Nikhil Prakash, Anja Reusch, Aruna Sankaranarayanan, Shun Shao, Alessandro Stolfo, Martin Tutek, Amir Zur, David Bau, and Yonatan Belinkov. Mib: A mechanistic interpretability benchmark, 2025. URL https://arxiv.org/abs/2504.13151.

Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models of self-supervised sequence models, 2023. URL https://arxiv.org/abs/2309.00941.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020. doi: 10.23915/distill.00024.001. URL https://distill.pub/2020/circuits/zoom-in/.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads, 2022. URL https://arxiv.org/abs/2209.11895.

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.

Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. The geometry of categorical and hierarchical concepts in large language models, 2025. URL https://arxiv.org/abs/2406.01506.

Gonçalo Paulo, Stepan Shabalin, and Nora Belrose. Transcoders beat sparse autoencoders for interpretability, 2025. URL https://arxiv.org/abs/2501.18823.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale, 2024. URL https://arxiv.org/abs/2406.17557.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah, and Neel Nanda. Improving dictionary learning with gated sparse autoencoders, 2024a. URL https://arxiv.org/abs/2404.16014.

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

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, 2019. URL https://arxiv.org/abs/1811.10154.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge, 2015. URL https://arxiv.org/abs/1409.0575.

Itay Safran and Ohad Shamir. Spurious local minima are common in two-layer relu neural networks, 2018. URL https://arxiv.org/abs/1712.08968.

Harshvardhan Saini, Yiming Tang, and Dianbo Liu. Bridging mechanistic interpretability and prompt engineering with gradient ascent for interpretable persona control, 2026. URL https://arxiv.org/abs/2601.02896.

Adam Scherlis, Yonatan Belinkov, Mariya Nezhurina, Nasim Rahaman, Omri Azulay, and Carles Richard. Polysemanticity and capacity in neural networks. *arXiv preprint arXiv:2210.01892*, 2022.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, October 2019. ISSN 1573-1405. doi: 10.1007/s11263-019-01228-7. URL http://dx.doi.org/10.1007/s11263-019-01228-7.

Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, Stella Biderman, Adria Garriga-Alonso, Arthur Conmy, Neel Nanda, Jessica Rumbelow, Martin Wattenberg, Nandi Schoots, Joseph Miller, Eric J. Michaud, Stephen Casper, Max Tegmark, William Saunders, David Bau, Eric Todd, Atticus Geiger, Mor Geva, Jesse Hoogland, Daniel Murfet, and Tom McGrath. Open problems in mechanistic interpretability, 2025. URL https://arxiv.org/abs/2501.16496.

Dong Shu, Xuansheng Wu, Haiyan Zhao, Daking Rai, Ziyu Yao, Ninghao Liu, and Mengnan Du. A survey on sparse autoencoders: Interpreting the internal mechanisms of large language models, 2025. URL https://arxiv.org/abs/2503.05613.

Elana Simon and James Zou. Interplm: Discovering interpretable features in protein language models via sparse autoencoders, 2024. URL https://arxiv.org/abs/2412.12101.

Daniel A. Spielman, Huan Wang, and John Wright. Exact recovery of sparsely-used dictionaries, 2012. URL https://arxiv.org/abs/1206.5882.

Ruoyu Sun and Zhi-Quan Luo. Guaranteed matrix completion via non-convex factorization. *IEEE Transactions on Information Theory*, 62(11):6535–6579, November 2016. ISSN 1557-9654. doi: 10.1109/tit.2016.2598574. URL http://dx.doi.org/10.1109/TIT.2016.2598574.

Yiming Tang and Bin Dong. Demonstration notebook: Finding the most suited in-context learning example from interactions, 2024. URL https://arxiv.org/abs/2406.10878.

Yiming Tang, Arash Lagzian, Srinivas Anumasa, Qiran Zou, Yingtao Zhu, Ye Zhang, Trang Nguyen, Yih-Chung Tham, Ehsan Adeli, Ching-Yu Cheng, Yilun Du, and Dianbo Liu. Human-like content analysis for generative ai with language-grounded sparse encoders, 2025a. URL https://arxiv.org/abs/2508.18236.

Yiming Tang, Abhijeet Sinha, and Dianbo Liu. How does my model fail? automatic identification and interpretation of physical plausibility failure modes with matryoshka transcoders, 2025b. URL https://arxiv.org/abs/2511.10094.

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

Viswanathan Visweswaran and CA Floudast. A global optimization algorithm (gop) for certain classes of nonconvex nlps—ii. application of theory and test problems. *Computers & chemical engineering*, 14(12): 1419–1434, 1990.

Miles Wang, Tom Dupré la Tour, Olivia Watkins, Alex Makelov, Ryan A. Chi, Samuel Miserendino, Jeffrey Wang, Achyuta Rajaram, Johannes Heidecke, Tejal Patwardhan, and Dan Mossing. Persona features control emergent misalignment, 2025. URL https://arxiv.org/abs/2506.19823.

Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. Axbench: Steering llms? even simple baselines outperform sparse autoencoders, 2025. URL https://arxiv.org/abs/2501.17148.

Haiyan Zhao, Zirui He, Yiming Tang, Fan Yang, Ali Payani, Dianbo Liu, and Mengnan Du. Rep2text: Decoding full text from a single llm token representation, 2026. URL https://arxiv.org/abs/2511.06571.

Qiran Zou, Hou Hei Lam, Wenhao Zhao, Yiming Tang, Tingting Chen, Samson Yu, Tianyi Zhang, Chang Liu, Xiangyang Ji, and Dianbo Liu. Fml-bench: Benchmarking machine learning agents for scientific research, 2026. URL https://arxiv.org/abs/2510.10472.

A Taxonomy for Sparse Dictionary Learning in Mechanistic Interpretability

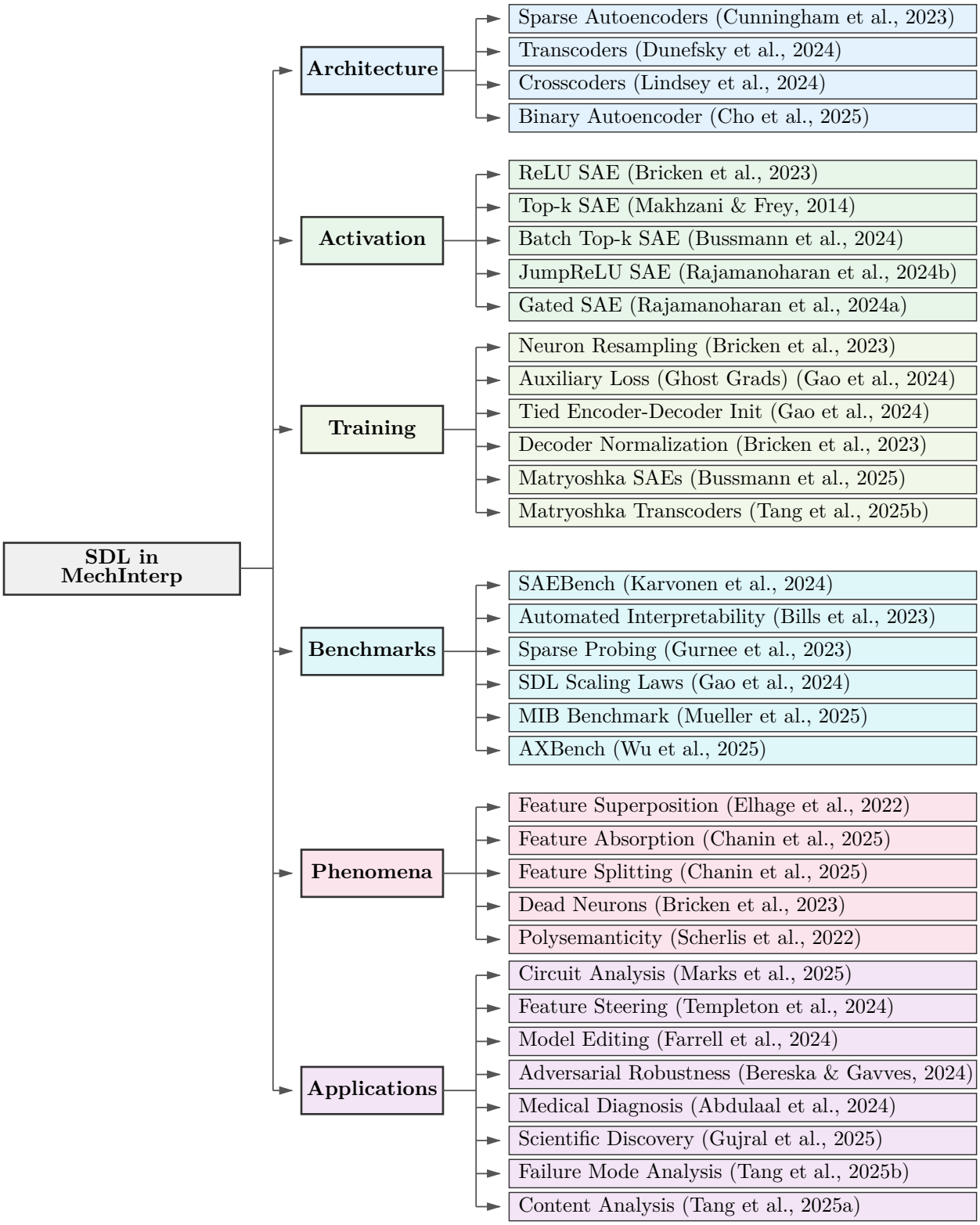

Figure 8: Hierarchical taxonomy of Sparse Dictionary Learning research in Mechanistic Interpretability.

B Notations

Notation	Description
Distributions and Indexing	
$\mathcal{D}$	Distribution over inputs
$\mathcal{X}$	Input space
s	Sample drawn from $\mathcal{D}$
$\mathbf{e}_d$	Standard basis vector (1 in position d, 0 elsewhere)
Representations and Dimensions	
$\mathbf{x}(s)$	Ground-truth features for input $s \sim \mathcal{D}$
$\mathbf{x}_p(s)$	Input representation to SDL model (observed)
$\mathbf{x}_r(s)$	Target/output representation for SDL model
$\mathbf{x}_q(s)$	Latent activations in SDL bottleneck
n	Number of ground-truth features (dimension of $\mathbf{x}$)
n_p	Dimension of input representation $\mathbf{x}_p$
n_r	Dimension of target representation $\mathbf{x}_r$
n_q	Dimension of latent space $\mathbf{x}_q$
SDL Architecture	
W_E	Encoder matrix ($n_q \times n_p$)
W_D	Decoder matrix ($n_r \times n_q$)
$\sigma(\cdot)$	Sparsity-inducing activation function
$\mathbf{w}_E^i$	i-th row of encoder W_E (encodes to neuron i)
$\mathbf{w}_D^i$	i-th column of decoder W_D (decodes from neuron i)
Linear Representation Hypothesis	
W_p	Feature matrix for $\mathbf{x}_p$ ($n_p \times n$)
W_r	Feature matrix for $\mathbf{x}_r$ ($n_r \times n$)
$\mathbf{w}_p^d$	d-th column of W_p (feature direction for feature d)
$\mathbf{w}_r^d$	d-th column of W_r (feature direction for feature d)
W_p^{true}	Ground-truth feature matrix (Linear Representation Bench)
W_p^{learned}	Learned feature matrix ($W_E^\top$)
Sparsity and Interference	
S	Sparsity level: $\Pr(\mathbf{x}_i(s) = 0) \geq S$
M	Maximum interference: $\max_{i \neq j} \langle W_p[:, i], W_p[:, j] \rangle$
M_d	Weight for feature d: $\Pr(\mathbf{x}(s) = x_d(s)\mathbf{e}_d) \cdot \mathbb{E}[x_d(s)^2 \mid x_d(s) > 0]$
Feature Anchoring	
k	Number of anchor features
$\tilde{\mathbf{w}}_p^{(i)}, \tilde{\mathbf{w}}_r^{(i)}$	i-th anchor feature pair
λ_{anchor}	Anchoring loss weight
$\mathcal{K}$	Set of indices for selected anchor features
Loss Functions	
$\mathcal{L}_{\text{SDL}}$	Standard SDL reconstruction loss
$\tilde{\mathcal{L}}_{\text{SDL}}$	Approximate SDL loss (extreme sparsity regime)
$\mathcal{L}_{\text{anchor}}$	Feature anchoring penalty
$\mathcal{L}_{\text{SDL-FA}}$	Anchored SDL objective: $\mathcal{L}_{\text{SDL}} + \lambda_{\text{anchor}} \mathcal{L}_{\text{anchor}}$
Activation Patterns	
$\mathcal{P} = (\mathcal{F}_1, \ldots, \mathcal{F}_N)$	Activation pattern
$\mathcal{F}_i$	Set of ground-truth features activating neuron i
$\mathcal{A}(d)$	Set of neurons activated by feature d
$\Omega_{\mathcal{A}}$	Activation pattern region (fixed activation structure)
Evaluation Metrics	
$\mathbf{M}_{\text{GT}}$	GT Recovery: fraction of features with $> \tau$ alignment
$\mathbf{M}_{\text{IP}}$	Maximum Inner Product: mean best alignment per feature
τ	Threshold for GT Recovery (typically $\tau = 0.9$ or 0.95)

C The Linear Representation Bench

The Linear Representation Bench is a synthetic benchmark designed to precisely instantiate the Representation Assumptions (Assumption 2.1) with fully accessible ground-truth features (See Figure 9 for a visualization of the benchmark). This enables rigorous evaluation of SDL methods under controlled conditions where feature recovery can be directly measured.

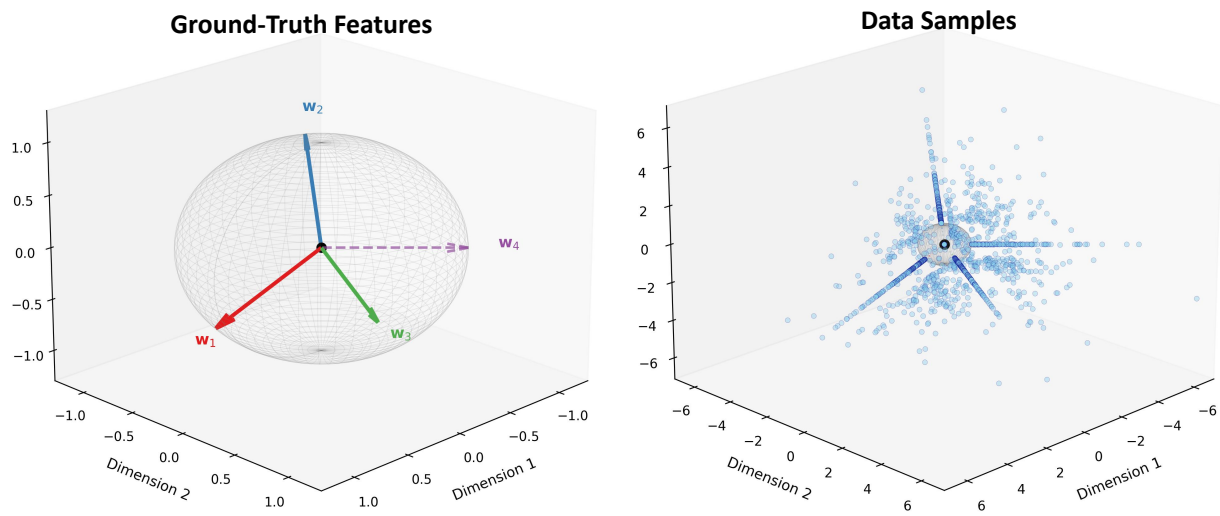

Figure 9: **Visualization of the Linear Representation Bench.** The figure illustrates a $D = 3$ dimensional representation space generated using $N = 4$ feature directions ($\mathbf{W}_p$).

C.1 Ground-Truth Feature Matrix Generation

We construct the feature matrix $\mathbf{W}_p \in \mathbb{R}^{n_p \times n}$ to satisfy the representation assumptions (Assumption 2.1).

Initialization. Initialize $\mathbf{W}_p$ with random Gaussian entries and normalize each column to unit ℓ_2-norm:

$$\mathbf{w}_p^d \leftarrow \frac{\mathbf{w}_p^d}{\|\mathbf{w}_p^d\|_2}, \quad \forall d \in [n] \tag{27}$$

Interference Minimization. We minimize pairwise interference via projected gradient descent. Define the soft-thresholded interference loss:

$$\mathcal{L}_{\text{int}}(\mathbf{W}_p) = \sum_{i \neq j} \left[\max\left(0, \langle \mathbf{w}_p^i, \mathbf{w}_p^j \rangle - (M - \epsilon)\right)\right]^2 + \lambda \sum_{i \neq j} \left[\max\left(0, \langle \mathbf{w}_p^i, \mathbf{w}_p^j \rangle\right)\right]^2 \tag{28}$$

where $\epsilon > 0$ is a tolerance margin and $\lambda > 0$ is a small regularization weight encouraging negative interference.

At each iteration, we compute the gradient with respect to $\mathbf{W}_p$, perform a gradient step, and project back to the unit sphere:

$$\mathbf{W}_p \leftarrow \mathbf{W}_p - \eta \nabla_{\mathbf{W}_p} \mathcal{L}_{\text{int}}, \quad \mathbf{w}_p^d \leftarrow \frac{\mathbf{w}_p^d}{\|\mathbf{w}_p^d\|_2}, \quad \forall d \in [n] \tag{29}$$

This procedure continues until the maximum interference satisfies $\max_{i \neq j} \langle \mathbf{w}_p^i, \mathbf{w}_p^j \rangle \leq M$.

C.2 Sparse Coefficient Generation

For each sample $i \in [N]$, we generate sparse ground-truth features $\mathbf{x}^{(i)} \in \mathbb{R}_+^n$ as follows:

Sparsity Mask. Each feature $d \in [n]$ is independently activated with probability $(1 - \mathcal{S})$:

$$m_d^{(i)} \sim \text{Bernoulli}(1 - \mathcal{S}) \tag{30}$$

Feature Magnitudes. Active features follow a shifted exponential distribution:

$$x_d^{(i)} = m_d^{(i)} \cdot (c_{\min} + \text{Exp}(\beta)) \tag{31}$$

where $c_{\min} \geq 0$ is a minimum activation threshold and $\beta > 0$ is the scale parameter.

This construction ensures:

- **Non-negativity:** $x_d^{(i)} \geq 0$ for all d, i
- **Sparsity:** $\Pr(x_d^{(i)} = 0) = \mathcal{S}$ for all d
- **Independence:** Features $\{x_d^{(i)}\}_{d=1}^n$ are mutually independent

C.3 Data Synthesis

The observed representations are computed via linear combination:

$$\mathbf{x}_p^{(i)} = \mathbf{W}_p \mathbf{x}^{(i)} = \sum_{d=1}^{n} x_d^{(i)} \mathbf{w}_p^d \tag{32}$$

By construction, this dataset exactly satisfies Assumptions 2.1, enabling evaluation of SDL methods with complete knowledge of ground-truth features.

C.4 Default Configuration

Table 3 lists the default parameters used in our experiments.

Table 3: Default configuration for the Linear Representation Bench.

Parameter	**Symbol**	**Default Value**
Number of features	n	1000
Representation dimension	n_p	768
Superposition ratio	n/n_p	1.30×
Number of samples	N	100,000
Sparsity level	$\mathcal{S}$	0.99
Maximum interference	M	0.1

This controlled setting enables direct measurement of feature recovery metrics with ground-truth access, complementing evaluations on real neural network representations where true features are unknown.

D Related Works

D.1 Mechanistic Interpretability

Interpretability is crucial for deploying AI in high-stakes domains such as medical diagnosis and financial modeling, where understanding model decisions is essential for safety and trust (Simon & Zou, 2024; Abdulaal et al., 2024; Zhao et al., 2026). Traditional approaches include interpretability-by-design methods like Concept Bottleneck Models (Koh et al., 2020) and decision trees (Mienye & Jere, 2024). Concept Activation Vectors (Kim et al., 2018) extend this by identifying human-defined concept directions in neural network representation spaces. Post-hoc explanation methods like GradCAM (Selvaraju et al., 2019) and SHAP (Lundberg & Lee, 2017) provide local explanations without modifying the model architecture. Mechanistic interpretability (Sharkey et al., 2025; Bereska & Gavves, 2024) aims to reverse-engineer neural networks by understanding their internal computational mechanisms. SAEs (Shu et al., 2025) and related dictionary learning methods (Tang et al., 2025a; Dunefsky et al., 2024) decompose neural activations into sparse, interpretable features. Circuit analysis (Olah et al., 2020; Olsson et al., 2022) investigates how these features compose into algorithms.

D.2 Sparse Dictionary Learning

Sparse dictionary learning has a rich history predating its application to mechanistic interpretability. K-SVD (Aharon et al., 2006) established foundational methods for learning overcomplete dictionaries, while theoretical work in compressed sensing (Donoho, 2006) characterized recovery conditions, with Spielman et al. (2012) providing polynomial-time algorithms for exact reconstruction under sparsity assumptions. Safran & Shamir (2018) demonstrated that spurious local minima are common even in simple two-layer ReLU networks, highlighting optimization challenges that persist in modern applications. Recent work has adapted these principles to mechanistic interpretability. SAEs (Cunningham et al., 2023) apply dictionary learning to language model activations, with LLMs widely utilized to analyze discovered features (Luo et al., 2024a; Luo et al.; Tang & Dong, 2024; Zou et al., 2026). Various variants are further developed, including transcoders (Dunefsky et al., 2024), crosscoders (Gao et al., 2024), Matryoshka SAEs (Bussmann et al., 2025), and hybrid approaches like Language-Grounded Sparse Encoders (Tang et al., 2025a). These methods have found applications beyond language models, including protein structure analysis (Simon & Zou, 2024; Gujral et al., 2025), medical imaging (Abdulaal et al., 2024), model evaluation (Tang et al., 2025b), prompt engineering (Saini et al., 2026), board game analysis (Karvonen et al., 2024), and fMRI data analysis Mao et al. (2025).

D.3 Biconvex Optimization

Biconvex optimization studies problems where the objective is convex in each variable block when the other is fixed. Gorski et al. (2007) provide a comprehensive survey establishing theoretical foundations and algorithmic approaches, while Visweswaran & Floudast (1990) develop the GOP algorithm providing global optimality guarantees via branch-and-bound. Matrix factorization problems exhibit similar bilinear structure; Lee et al. (2009) introduce non-negative matrix factorization with multiplicative updates, and subsequent work establishes landscape properties, with Ge et al. (2018) proving matrix completion has no spurious local minima and Sun & Luo (2016) providing convergence guarantees for alternating minimization. More broadly, Jain & Kar (2017) survey non-convex optimization in machine learning, while Safran & Shamir (2018) demonstrate that spurious local minima are common in two-layer ReLU networks. Our work bridges these optimization-theoretic foundations with mechanistic interpretability by proving SDL exhibits biconvex structure, enabling the application of established algorithms and analysis techniques to this emerging field.

E Discussion

E.1 Feature Anchoring and Evaluation Fairness.

A natural concern is whether feature anchoring constitutes an unfair advantage given its partial access to ground-truth directions. We emphasize that evaluation is performed exclusively on non-anchored features (Table 1), so reported metrics directly measure unseen feature recovery. Notably, anchoring consistently improves recovery across the entire feature space—not just features close to anchored directions—suggesting an emerging alignment effect: constraining a small subset of features to known semantic directions reduces the degrees of freedom in the solution space, making spurious partial minima less reachable for the remaining features and encouraging broader monosemantic organization.

E.2 ReLU and JumpReLU SAEs Underperforms Under Feature Anchoring.

Despite anchoring, ReLU and JumpReLU SAEs achieve 0% GT Recovery, though this reflects threshold effects rather than complete failure: best-match scores typically fall in 0.2–0.4 on the Linear Representation Bench and 0.6–0.8 on CLIP embeddings. We also note that while these methods do discover seemingly interpretable patterns in practice, it remains unclear whether such patterns genuinely correspond to monosemantic directions under the Linear Representation Hypothesis—the expressive power of natural language may allow polysemantic features to appear interpretable upon inspection. This points to a broader open question: for large-scale models, what constitutes a ground-truth feature, and how should it be rigorously evaluated? The absence of a consensus benchmark for feature-level ground truth remains an important unsolved problem in mechanistic interpretability.

E.3 What the Biconvexity Structure Indicates?

The piecewise biconvex structure of SDL optimization (Theorem 3.2) serves two complementary roles. Theoretically, it provides the analytical foundation for our landscape characterization: within each activation pattern region, the loss is convex in each variable block separately, enabling rigorous analysis of global minima, spurious partial minima, and feature absorption as structural properties rather than empirical observations. Algorithmically, it bridges SDL with a rich body of biconvex optimization literature, opening the door to methods such as GOP and alternating convex search that offer systematic escape from spurious partial minima with theoretical guarantees—directions we identify as promising future work.

E.4 SDL Method Recommendations for Practitioners.

Based on our theoretical analysis and experimental results, we offer the following practical guidance. TopK-family SAEs (TopK, BatchTopK, Matryoshka) are the most reliable choices, consistently achieving 84–90% ground-truth recovery on the Linear Representation Bench, and we recommend combining them with feature anchoring for further gains. ReLU and JumpReLU SAEs should be avoided as primary methods, as they consistently fail to recover ground-truth features across all evaluated settings.

E.5 Relationship to Concept Activation Vectors.

Our subpopulation mean embeddings used as anchors bear surface similarity to Concept Activation Vectors (CAVs) (Kim et al., 2018), which also identify concept directions via class-mean representations. However, the two serve fundamentally different purposes: CAVs are a post-hoc analysis tool for probing whether a concept influences a model's predictions, whereas feature anchoring uses these directions to constrain the SDL training process itself. In other words, CAVs are used to interpret a fixed model, while feature anchoring uses similar geometric information to guide optimization toward more interpretable solutions.

E.6 On Theoretical Assumptions.

Our analysis operates under idealized assumptions—exact linear decomposition, mutual feature independence, and extreme sparsity—that may not hold perfectly in real neural networks. We acknowledge these as

deliberate simplifications that make the optimization landscape tractable for rigorous analysis. Empirically, the polysemantic features consistently observed without feature anchoring on real CLIP embeddings (Table 2) suggest that our theoretical results on spurious partial minima persist beyond our idealized assumptions. As the first unified theoretical framework for SDL in mechanistic interpretability, we view our work as a foundation that establishes the core landscape properties, and we encourage future work to progressively relax these assumptions—extending results to approximate linearity, correlated features, and moderate sparsity.

F Limitations

While our theoretical framework provides valuable insights into SDL methods, several limitations exists:

- **Assumption Violations.** Our analysis relies on the Representation Assumptions (Assumption 2.1), which may not hold perfectly in real-world neural networks. In particular, there exists features that are not one-dimentional linear (Engels et al., 2025) .
- **Extreme Sparsity.** Several key results, including Theorem 3.1, rely on the extreme sparsity assumption ($1-S < \frac{1}{n}$). The bounds may degrade substantially for moderate sparsity levels commonly observed in practice.
- **Feature Independence.** We assume mutual independence among ground-truth features, but real-world concepts often exhibit correlations that our analysis does not capture.
- **Convergence Guarantees.** We characterize global and partial minima of the optimization landscape but do not provide guarantees on whether gradient-based methods converge to global minima.
- **Anchor Availability.** Feature anchoring requires access to known semantic directions, which may not always be available. The quality and coverage of anchors significantly impact performance.

G Future Works

Our theoretical framework opens several avenues for future research:

- **Global Optimization via GOP.** Having established that SDL is piecewise biconvex (Theorem 3.2), applying Global Optimization for biconvex Problems (GOP) (Gorski et al., 2007) could provide certificates of global optimality and systematically escape spurious partial minima.
- **Alternating Convex Search.** Alternating convex search (ACS) (Gorski et al., 2007) directly exploits biconvex structure by alternately solving convex subproblems for W_D and W_E. This may offer faster convergence and better solutions than standard gradient descent.
- **Convergence and Sample Complexity.** Building on our landscape characterization, future work should establish convergence rates for gradient descent and sample complexity bounds.
- **Moderate Sparsity Analysis.** Extending results beyond the $S \to 1$ regime to handle feature co-activation would broaden practical applicability.
- **Classical Dictionary Learning.** Adapting established algorithms like K-SVD (Aharon et al., 2006) and online dictionary learning (Mairal et al., 2009) to the SDL setting is promising.
- **Automatic Anchor Discovery.** Developing methods to automatically discover high-quality anchors without external supervision would make feature anchoring more practical.

H Ablation Studies

Our theoretical analysis reveals that SDL optimization becomes increasingly underdetermined as the number of ground-truth features n grows relative to the representation dimension n_p (Theorem 3.4). To comprehensively validate our theoretical framework and assess the robustness of feature anchoring across different conditions, we conduct extensive ablation studies on the Linear Representation Bench, systematically varying superposition ratio, interference level, sparsity, and activation function parameters.

Effect of Superposition Ratio. We first examine how feature anchoring performs under varying degrees of superposition by training ReLU SAEs with $n_q = 16,000$ latent dimensions while varying the number of ground-truth features $n \in \{800, 900, 1000, 1100, 1200, 1300, 1400\}$. The representation dimension is fixed at $n_p = n_r = 768$, yielding superposition ratios from $1.04\times$ to $1.82\times$. For feature anchoring, we randomly select $k = 100$ ground-truth features as anchors with $\lambda_{\text{anchor}} = 0.1$.

As shown in Table 4, standard SAE training achieves 0% GT Recovery across all feature counts, demonstrating the severity of the underdetermined optimization problem. With feature anchoring, GT Recovery reaches 100% at $n = 800$ and remains high (98.89%) at $n = 900$, confirming that anchoring a small subset of features (12.5% and 11.1% respectively) provides sufficient constraint to guide optimization toward the global minimum. As superposition increases, GT Recovery gradually decreases (85.50% at $1.30\times$, 58.09% at $1.43\times$, 30.58% at $1.56\times$), aligning with our theoretical expectation that higher superposition expands the space of spurious solutions. Notably, feature anchoring consistently improves Maximum Inner Product across all settings, indicating better feature quality even when full recovery is not achieved.

Table 4: Feature recovery under varying superposition ratios. We fix $n_p = n_r = 768$ and vary n from 800 to 1400. Feature anchoring uses $k = 100$ randomly selected anchors with $\lambda_{\text{anchor}} = 0.1$.

Method	Num Features	GT Recovery ↑	Max Inner Product ↑
SAE	800	0.00%	0.361
+ Feature Anchoring	800	**100.00%**	**0.436**
SAE	900	0.00%	0.298
+ Feature Anchoring	900	**98.89%**	**0.345**
SAE	1000	0.00%	0.254
+ Feature Anchoring	1000	**85.50%**	**0.292**
SAE	1100	0.00%	0.223
+ Feature Anchoring	1100	**58.09%**	**0.258**
SAE	1200	0.00%	0.203
+ Feature Anchoring	1200	**30.58%**	**0.233**
SAE	1300	0.00%	0.192
+ Feature Anchoring	1300	**15.23%**	**0.216**
SAE	1400	0.00%	0.181
+ Feature Anchoring	1400	**8.57%**	**0.207**

Effect of Maximum Interference. To examine how feature anchoring performs under different interference levels, we fix $n = 1000$ features in $n_p = n_r = 768$ dimensions and vary the maximum interference $M \in \{0.05, 0.1, 0.2, 0.5\}$ during ground-truth feature matrix generation. Recall that $M := \max_{i \neq j} \langle W_p[:, i], W_p[:, j] \rangle$ quantifies the maximum dot product between distinct feature directions, characterizing the degree of feature overlap in the representation space.

Table 5 presents results across different interference levels. For TopK, BatchTopK, and Matryoshka SAEs, feature anchoring consistently improves GT Recovery across all interference values, with improvements ranging from 2-7 percentage points. For instance, at $M = 0.5$, BatchTopK SAE improves from 83.7% to 89.6% with anchoring. ReLU and JumpReLU SAEs, which struggle with feature recovery even with anchoring

(0% GT Recovery), still show improved Maximum Inner Product (e.g., ReLU at $M = 0.2$: $0.204 \rightarrow 0.245$), indicating better feature alignment. Interestingly, performance degrades slightly at very low interference ($M = 0.05$) compared to moderate interference ($M = 0.1, 0.2$), likely because extremely low interference creates near-orthogonal features that are easier to recover without additional constraints, making the anchoring less critical.

Table 5: Feature recovery under varying maximum interference M. All experiments use $n = 1000$ features with $n_p = n_r = 768$ and $k = 100$ anchors.

Method	**Max Interference**	**GT Recovery ↑**	**Max Inner Product ↑**
Low Interference ($M = 0.05$)			
ReLU SAE	0.05	0.0%	0.205
+ Feature Anchoring	0.05	0.0%	**0.246**
JumpReLU SAE	0.05	0.0%	0.239
+ Feature Anchoring	0.05	0.0%	**0.326**
TopK SAE	0.05	84.3%	0.982
+ Feature Anchoring	0.05	**87.3%**	**0.985**
BatchTopK SAE	0.05	86.1%	0.984
+ Feature Anchoring	0.05	**87.5%**	**0.986**
Matryoshka SAE	0.05	83.8%	0.982
+ Feature Anchoring	0.05	**85.8%**	**0.985**
Moderate Interference ($M = 0.2$)			
ReLU SAE	0.2	0.0%	0.204
+ Feature Anchoring	0.2	0.0%	**0.245**
JumpReLU SAE	0.2	0.0%	0.237
+ Feature Anchoring	0.2	0.0%	**0.333**
TopK SAE	0.2	85.2%	0.984
+ Feature Anchoring	0.2	**86.8%**	**0.986**
BatchTopK SAE	0.2	84.7%	0.981
+ Feature Anchoring	0.2	**90.7%**	**0.989**
Matryoshka SAE	0.2	84.7%	0.983
+ Feature Anchoring	0.2	**87.6%**	**0.986**
High Interference ($M = 0.5$)			
ReLU SAE	0.5	0.0%	0.203
+ Feature Anchoring	0.5	0.0%	**0.243**
JumpReLU SAE	0.5	0.0%	0.237
+ Feature Anchoring	0.5	0.0%	**0.325**
TopK SAE	0.5	84.2%	0.982
+ Feature Anchoring	0.5	**88.2%**	**0.987**
BatchTopK SAE	0.5	83.7%	0.981
+ Feature Anchoring	0.5	**89.6%**	**0.987**
Matryoshka SAE	0.5	83.9%	0.982
+ Feature Anchoring	0.5	**86.4%**	**0.985**

Effect of Feature Sparsity. We investigate how feature sparsity S affects SDL performance by varying $S \in \{0.005, 0.01, 0.05, 0.1\}$ where S denotes the probability that each feature is inactive. Lower S values correspond to denser activation patterns, while higher S approaches the extreme sparsity regime analyzed in our theoretical results.

Table 6 reveals several key insights. At very low sparsity ($S = 0.005$), where features co-activate frequently, feature anchoring provides the most dramatic improvements. JumpReLU SAE improves from 11.8% to 74.8% GT Recovery, and all TopK-family methods show 2-6 percentage point gains. This validates that anchoring is particularly valuable when the extreme sparsity assumption is violated and feature co-occurrence complicates

optimization. At moderate sparsity ($S = 0.05$), performance drops significantly across all methods, as increased co-activation creates more complex interference patterns. At high sparsity ($S = 0.1$), approaching our theoretical regime, TopK-family methods achieve near-perfect recovery (98.8%-99.7%), with anchoring providing marginal improvements. Interestingly, Matryoshka SAE benefits least from anchoring at extreme sparsity, likely because its multi-scale architecture already provides sufficient constraints to recover features.

Table 6: Feature recovery under varying sparsity levels S. All experiments use $n = 1000$ features with $n_p = n_r = 768$ and $k = 100$ anchors.

Method	**Sparsity**	**GT Recovery ↑**	**Max Inner Product ↑**
Very Low Sparsity ($S = 0.005$)			
ReLU SAE	0.005	0.0%	0.231
+ Feature Anchoring	0.005	0.0%	**0.326**
JumpReLU SAE	0.005	11.8%	0.890
+ Feature Anchoring	0.005	**74.8%**	**0.970**
TopK SAE	0.005	82.2%	0.978
+ Feature Anchoring	0.005	**84.3%**	**0.981**
BatchTopK SAE	0.005	80.5%	0.976
+ Feature Anchoring	0.005	**83.2%**	**0.980**
Matryoshka SAE	0.005	79.3%	0.974
+ Feature Anchoring	0.005	**86.1%**	**0.983**
Moderate Sparsity ($S = 0.01$)			
ReLU SAE	0.01	0.00%	0.205
+ Feature Anchoring	0.01	0.00%	**0.246**
JumpReLU SAE	0.01	0.00%	0.237
+ Feature Anchoring	0.01	0.00%	**0.333**
TopK SAE	0.01	84.90%	0.983
+ Feature Anchoring	0.01	**87.63%**	**0.986**
BatchTopK SAE	0.01	84.80%	0.981
+ Feature Anchoring	0.01	**89.38%**	**0.988**
Matryoshka SAE	0.01	83.70%	0.982
+ Feature Anchoring	0.01	**87.32%**	**0.985**
High Sparsity ($S = 0.1$)			
ReLU SAE	0.1	0.0%	0.186
+ Feature Anchoring	0.1	0.0%	**0.191**
JumpReLU SAE	0.1	0.0%	0.191
+ Feature Anchoring	0.1	0.0%	**0.193**
TopK SAE	0.1	98.8%	0.981
+ Feature Anchoring	0.1	98.4%	0.979
BatchTopK SAE	0.1	95.6%	0.975
+ Feature Anchoring	0.1	**96.2%**	**0.976**
Matryoshka SAE	0.1	99.7%	0.991
+ Feature Anchoring	0.1	99.7%	0.991

Effect of TopK Sparsity Parameter. Finally, we examine how the sparsity-inducing parameter k in TopK activation affects feature recovery. We train TopK SAEs with $k \in \{32, 64, 128\}$, corresponding to average activation rates of 0.2%, 0.4%, and 0.8% of the $n_q = 16,000$ latent dimensions.

As shown in Table 7, both standard and anchored TopK SAEs achieve perfect (100%) GT Recovery at $k = 32$, where extreme sparsity closely matches our theoretical assumptions. At $k = 64$, GT Recovery remains high (93.4%-93.9%) with minimal difference between standard and anchored training, indicating that moderate sparsity provides sufficient constraint for feature recovery without additional anchoring. At $k = 128$, where activation density increases, feature anchoring becomes beneficial again, improving GT Recovery from 86.7%

to 88.0%. The Maximum Inner Product remains consistently high (> 0.985) across all settings, confirming that TopK activation is generally effective for feature recovery, with anchoring providing incremental benefits at higher k values where the optimization becomes more challenging.

Table 7: Feature recovery under varying TopK sparsity parameter k. All experiments use $n = 1000$ features with $n_p = n_r = 768$, $n_q = 16,000$, and $k = 100$ anchors.

Method	**TopK Parameter k**	**GT Recovery ↑**	**Max Inner Product ↑**
TopK SAE	32	100.0%	0.999
+ Feature Anchoring	32	**100.0%**	**0.999**
TopK SAE	64	93.9%	0.992
+ Feature Anchoring	64	93.4%	0.991
TopK SAE	128	86.7%	0.986
+ Feature Anchoring	128	**88.0%**	**0.987**

Theoretical Interpretation. These comprehensive ablation studies validate our theoretical framework across diverse conditions. The consistent failure of ReLU and JumpReLU SAEs without sufficient sparsity constraints corroborates Theorem 3.4: the underdetermined solution space admits infinitely many configurations achieving low reconstruction loss without recovering interpretable features. Feature anchoring addresses this by constraining encoder-decoder pairs, reducing degrees of freedom and steering optimization away from spurious partial minima (Theorem 3.7). The varying effectiveness across interference levels, sparsity regimes, and activation functions reflects the fundamental trade-off in our assumptions: more severe feature compression (higher M), denser activations (lower S), or insufficient activation sparsity (higher k) create more complex optimization landscapes where even anchored methods struggle to disentangle all features completely.

I Qualitative Examples

Beyond quantitative metrics, we provide qualitative evidence demonstrating how feature anchoring improves feature monosemanticity. Figure 10 shows features learned by Matryoshka SAE with feature anchoring on CLIP, while Figure 11 shows features from the same architecture trained without feature anchoring.

With Feature Anchoring. As shown in Figure 10, the learned features exhibit clear monosemanticity: each feature responds to a single, well-defined visual concept. The "African Grey" feature activates exclusively on African Grey parrots, the "American Black Bear" feature captures only black bears, and the "Digital Clock" feature responds specifically to digital time displays. This monosemantic behavior aligns with our theoretical prediction that feature anchoring reduces the underdetermined nature of SDL optimization (Theorem 3.4).

Without Feature Anchoring. In contrast, Figure 11 illustrates the polysemanticity that emerges without anchoring. Features respond to multiple unrelated concepts: one activates on "Various Boxes" (dishwashers, file cabinets, chests), another on "Various Screens" (slot machines, scoreboards, televisions). While achieving low reconstruction loss, these features fail to capture semantically meaningful concepts—exactly the spurious partial minima characterized in Theorem 3.6.

Figure 10: **Features learned with feature anchoring exhibit monosemanticity.** Each row shows the top-activating images for a single feature from Matryoshka SAE trained on CLIP with feature anchoring.

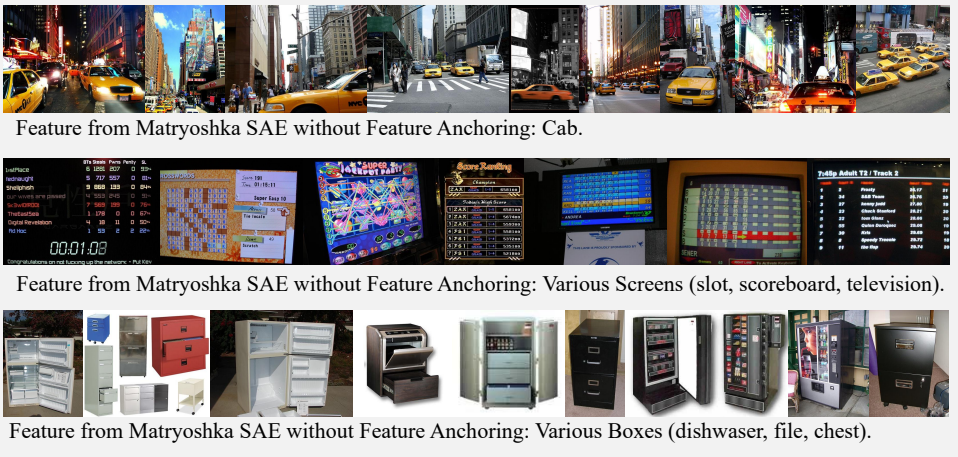

Figure 11: **Features learned without feature anchoring exhibit polysemanticity.** Each row shows the top-activating images for a single feature from Matryoshka SAE trained without feature anchoring.

J Proofs of the Theorems

J.1 Proof of Theorem 3.1

Proof. Let $C = \sup_{s\sim\mathcal{D}} \|\mathbf{x}_r(s) - W_D\sigma(W_E\mathbf{x}_p(s))\|_2^2 < \infty$ denote the uniform upper bound on the squared reconstruction error. The loss decomposes over the number of active features:

$$\mathcal{L}_{\text{SDL}} = \sum_{m=0}^{n} \Pr(\|\mathbf{x}(s)\|_0 = m) \cdot \mathbb{E}\big[\|\mathbf{x}_r(s) - W_D\sigma(W_E\mathbf{x}_p(s))\|^2 \mid \|\mathbf{x}(s)\|_0 = m\big]. \tag{33}$$

The $m = 0$ term contributes zero. For $m = 1$, exactly one feature d is active, giving $\mathbf{x}_r(s) = x_d(s)w_r^d$ and $\mathbf{x}_p(s) = x_d(s)w_p^d$, so the total contribution equals:

$$\sum_{d=1}^{n} M_d \left\|w_r^d - W_D\sigma(W_E w_p^d)\right\|^2 = \tilde{\mathcal{L}}_{\text{SDL}}(W_D, W_E), \tag{34}$$

where $M_d := \Pr(\mathbf{x}(s) = x_d(s)\mathbf{e}_d) \cdot \mathbb{E}[x_d(s)^2 \mid x_d(s) > 0]$. For $m \geq 2$, since features are independently activated with probability $(1 - S)$, a union bound over all pairs $i \neq j$ gives:

$$\Pr(\|\mathbf{x}(s)\|_0 \geq 2) \leq \sum_{i,j} \Pr(x_i > 0,\, x_j > 0) \leq n^2(1 - S)^2. \tag{35}$$

Since the reconstruction error is bounded by C, combining all cases yields:

$$\left|\mathcal{L}_{\text{SDL}} - \tilde{\mathcal{L}}_{\text{SDL}}\right| \leq n^2 C(1 - S)^2. \tag{36}$$

□

J.2 Proof of Theorem 3.2

Proof. **Preliminary observation.** By definition of $\Omega_{\mathcal{P}}$ (Definition 3.1), for any $W_E \in \Omega_{\mathcal{P}}$, the activation $z_d(W_E) := \sigma(W_E w_p^d)$ satisfies:

$$z_d(W_E)_i = \begin{cases} \langle w_E^i, w_p^d\rangle & \text{if } i \in \mathcal{F}_d \\ 0 & \text{otherwise,} \end{cases} \tag{37}$$

where $\mathcal{F}_d = \{i \in [n_q] : (\sigma(W_E w_p^d))_i > 0\}$. Hence $z_d(W_E)$ is linear in W_E within $\Omega_{\mathcal{P}}$.

Per-feature decomposition. Define $f_d(W_D, W_E) := \|w_r^d - W_D z_d(W_E)\|^2$. Then:

$$\tilde{\mathcal{L}}_{\text{SDL}}(W_D, W_E) = \sum_{d=1}^{n} M_d f_d(W_D, W_E), \quad M_d > 0. \tag{38}$$

Since positive linear combinations preserve biconvexity, it suffices to show each f_d is biconvex.

Convexity in W_D. For fixed $W_E \in \Omega_{\mathcal{P}}$, z_d is a fixed vector, so $f_d(W_D) = \|w_r^d - W_D z_d\|^2$ is a squared affine function of W_D, hence convex. Its Hessian $\nabla^2_{W_D} f_d = 2I_{n_r} \otimes (z_d z_d^\top) \succeq 0$ confirms this.

Convexity in W_E. For fixed W_D, linearity of $z_d(W_E)$ in W_E gives:

$$W_D z_d(W_E) = \sum_{i\in\mathcal{F}_d} \langle w_E^i, w_p^d\rangle w_D^i, \tag{39}$$

which is affine in W_E. Hence $f_d(W_E) = \|w_r^d - W_D z_d(W_E)\|^2$ is a squared affine function of W_E, which is convex. For each row w_E^i with $i \in \mathcal{F}_d$, the Hessian $\nabla^2_{w_E^i} f_d = 2\|w_D^i\|^2(w_p^d(w_p^d)^\top) \succeq 0$, and rows $i \notin \mathcal{F}_d$ do not affect f_d.

Conclusion. Each f_d is biconvex, so $\tilde{\mathcal{L}}_{\text{SDL}}$ is biconvex over $\mathbb{R}^{n_r\times n_q} \times \Omega_{\mathcal{P}}$. □

J.3 Proof of Theorem 3.3

Proof. We analyze $\tilde{\mathcal{L}}_{\text{SDL}}(W_D^*, W_E^*)$ by examining the reconstruction error for each feature $d \in [n]$ active in isolation.

The encoder output is $W_E^* w_p^d = W_p^\top w_p^d$ with k-th component $(W_E^* w_p^d)_k = \langle w_p^k, w_p^d \rangle$. By the unit-norm condition (Assumption 2.1), $(W_E^* w_p^d)_d = 1$. Define the set of neurons activated by feature d:

$$\mathcal{A}_d = \{k \in [n] : \sigma(W_E^* w_p^d)_k > 0\}. \tag{40}$$

Since $\sigma(z)_i \in \{0, z_i\}$, we have $\sigma(W_E^* w_p^d)_k > 0$ if and only if $\langle w_p^k, w_p^d \rangle > 0$. Therefore $d \in \mathcal{A}_d$, and for $k \neq d$:

$$\mathcal{A}_d \setminus \{d\} \subseteq \{k \in [n] : 0 < \langle w_p^k, w_p^d \rangle \leq M\}. \tag{41}$$

The reconstruction is $W_D^* \sigma(W_E^* w_p^d) = \sum_{k \in \mathcal{A}_d} \langle w_p^k, w_p^d \rangle w_r^k$, so the reconstruction error for feature d is:

$$w_r^d - W_D^* \sigma(W_E^* w_p^d) = w_r^d - \sum_{k \in \mathcal{A}_d} \langle w_p^k, w_p^d \rangle w_r^k = - \sum_{k \in \mathcal{A}_d \setminus \{d\}} \langle w_p^k, w_p^d \rangle w_r^k. \tag{42}$$

Applying the triangle inequality and using $\|w_r^k\|_2 = 1$ and $\langle w_p^k, w_p^d \rangle \leq M$ for $k \in \mathcal{A}_d \setminus \{d\}$:

$$\left\| w_r^d - W_D^* \sigma(W_E^* w_p^d) \right\|_2 \leq \sum_{k \in \mathcal{A}_d \setminus \{d\}} \langle w_p^k, w_p^d \rangle \leq M(K_d - 1), \tag{43}$$

where $K_d = |\mathcal{A}_d| \leq n$. Squaring and summing over all features:

$$\tilde{\mathcal{L}}_{\text{SDL}}(W_D^*, W_E^*) \leq M^2 \sum_{d=1}^{n} M_d (K_d - 1)^2 \leq n^2 M^2 \sum_{d=1}^{n} M_d, \tag{44}$$

where the second inequality uses $(K_d - 1)^2 \leq (n-1)^2 \leq n^2$. □

J.4 Proof of Theorem 3.4

Proof. We prove both directions of the equivalence.

Sufficient condition (⇒). Suppose $w_r^d = W_D \sigma(W_E w_p^d)$ for all $d \in [n]$. By definition of the approximate loss:

$$\begin{aligned} \tilde{\mathcal{L}}_{\text{SDL}}(W_D, W_E) &= \sum_{d=1}^{n} M_d \|w_r^d - W_D \sigma(W_E w_p^d)\|^2 && (45) \\ &= \sum_{d=1}^{n} M_d \|w_r^d - w_r^d\|^2 && (46) \\ &= 0 && (47) \end{aligned}$$

Necessary condition (⇐). Suppose $\tilde{\mathcal{L}}_{\text{SDL}}(W_D, W_E) = 0$. Then:

$$\sum_{d=1}^{n} M_d \|w_r^d - W_D \sigma(W_E w_p^d)\|^2 = 0 \tag{48}$$

Since $M_d > 0$ for all $d \in [n]$ (by definition, $M_d = \Pr(\mathbf{x}(s) = x_d(s) e_d) \cdot \mathbb{E}[x_d(s)^2 | x_d(s) > 0]$ where both factors are positive under Assumptions 2.1):

$$\|w_r^d - W_D \sigma(W_E w_p^d)\|^2 = 0 \quad \forall d \in [n] \tag{49}$$

Therefore:

$$w_r^d = W_D \sigma(W_E w_p^d) \quad \forall d \in [n] \tag{50}$$

□

J.5 Proof of Theorem 3.6

Proof. We construct a configuration (W_D^*, W_E^*) exhibiting the polysemanticity pattern and explicitly verify that both gradients vanish.

Step 1: Given encoder W_E^* from realizability.

Since the activation pattern $\mathcal{P} = (\mathcal{F}_1, \ldots, \mathcal{F}_{n_q})$ is realizable and forms a partition of $[n]$, by Definition 3.1, there exists an encoder $W_E^* \in \mathbb{R}^{n_q \times n_p}$ such that:

$$\forall i \in [n_q], \quad \mathcal{F}_i = \left\{ d \in [n] : \left(\sigma(W_E^* w_p^d)\right)_i > 0 \right\} \tag{51}$$

Fix this W_E^* and define the activation vectors:

$$z_d := \sigma(W_E^* w_p^d) \in \mathbb{R}^{n_q}, \quad d = 1, \ldots, n \tag{52}$$

Since $\mathcal{P}$ forms a partition, for each feature d there exists a unique neuron $i(d)$ such that $d \in \mathcal{F}_{i(d)}$, and:

$$z_d = \langle w_E^{*,i(d)}, w_p^d \rangle \cdot e_{i(d)} \tag{53}$$

That is, only neuron $i(d)$ activates for feature d.

Step 2: Construct optimal decoder W_D^*.

For each neuron $i \in [n_q]$, we construct the decoder column $W_D^{*,i}$ by minimizing the loss over features activating that neuron.

Dead neurons: If $\mathcal{F}_i = \emptyset$, set $W_D^{*,i} = 0$.

Active neurons: If $\mathcal{F}_i \neq \emptyset$, solve:

$$W_D^{*,i} = \arg\min_{W_D^i \in \mathbb{R}^{n_r}} \sum_{d \in \mathcal{F}_i} M_d \left\| w_r^d - W_D^i (z_d)_i \right\|^2 \tag{54}$$

This is a least squares problem with solution:

$$W_D^{*,i} = \frac{\sum_{d \in \mathcal{F}_i} M_d (z_d)_i w_r^d}{\sum_{d \in \mathcal{F}_i} M_d (z_d)_i^2} \tag{55}$$

Step 3: Verify $\nabla_{W_D} \tilde{\mathcal{L}}_{\mathbf{SDL}}(W_D^*, W_E^*) = 0$.

The loss is $\tilde{\mathcal{L}}_{\text{SDL}}(W_D, W_E) = \sum_{d=1}^n M_d \|w_r^d - W_D z_d\|^2$. The gradient with respect to column W_D^i is:

$$\nabla_{W_D^i} \tilde{\mathcal{L}}_{\text{SDL}} = -2 \sum_{d=1}^n M_d (w_r^d - W_D z_d)(z_d)_i \tag{56}$$

Since $\mathcal{P}$ is a partition, $(z_d)_i \neq 0$ only when $i = i(d)$, i.e., when $d \in \mathcal{F}_i$. Therefore:

$$\nabla_{W_D^i} \tilde{\mathcal{L}}_{\text{SDL}}(W_D^*, W_E^*) = -2 \sum_{d \in \mathcal{F}_i} M_d (w_r^d - W_D^{*,i} (z_d)_i)(z_d)_i \tag{57}$$

By construction (55), $W_D^{*,i}$ satisfies the normal equation:

$$\sum_{d \in \mathcal{F}_i} M_d (z_d)_i (w_r^d - W_D^{*,i} (z_d)_i) = 0 \tag{58}$$

Therefore $\nabla_{W_D^i} \tilde{\mathcal{L}}_{\text{SDL}}(W_D^*, W_E^*) = 0$ for all $i \in [n_q]$.

Step 4: Verify $\nabla_{W_E} \tilde{\mathcal{L}}_{\textbf{SDL}}(W_D^*, W_E^*) = 0$.

Within the activation pattern region $\Omega_{\mathcal{A}}$, for feature $d \in \mathcal{F}_i$, the activation is:

$$z_d = \langle w_E^i, w_p^d \rangle \cdot e_i \tag{59}$$

The gradient with respect to encoder row w_E^i is:

$$\nabla_{w_E^i} \tilde{\mathcal{L}}_{\text{SDL}} = \sum_{d=1}^{n} M_d \nabla_{w_E^i} \left\| w_r^d - W_D z_d \right\|^2 \tag{60}$$

Only features $d \in \mathcal{F}_i$ have non-zero contribution. For such d:

$$\frac{\partial z_d}{\partial w_E^i} = w_p^d \otimes e_i \tag{61}$$

where $\otimes$ denotes outer product. Therefore:

$$\nabla_{w_E^i} \left\| w_r^d - W_D z_d \right\|^2 = -2(w_r^d - W_D z_d)^\top W_D \frac{\partial z_d}{\partial w_E^i} \tag{62}$$

$$= -2(w_r^d - W_D^{*,i} \langle w_E^{*,i}, w_p^d \rangle)^\top W_D^{*,i} w_p^d \tag{63}$$

The total gradient is:

$$\nabla_{w_E^i} \tilde{\mathcal{L}}_{\text{SDL}} = -2 \sum_{d \in \mathcal{F}_i} M_d (w_r^d - W_D^{*,i} \langle w_E^{*,i}, w_p^d \rangle)^\top W_D^{*,i} w_p^d \tag{64}$$

$$= -2 (W_D^{*,i})^\top \sum_{d \in \mathcal{F}_i} M_d \langle w_E^{*,i}, w_p^d \rangle (w_r^d - W_D^{*,i} \langle w_E^{*,i}, w_p^d \rangle) \tag{65}$$

From the normal equation in Step 3 with $(z_d)_i = \langle w_E^{*,i}, w_p^d \rangle$:

$$\sum_{d \in \mathcal{F}_i} M_d \langle w_E^{*,i}, w_p^d \rangle (w_r^d - W_D^{*,i} \langle w_E^{*,i}, w_p^d \rangle) = 0 \tag{66}$$

Therefore:

$$\nabla_{w_E^i} \tilde{\mathcal{L}}_{\text{SDL}}(W_D^*, W_E^*) = -2 (W_D^{*,i})^\top \cdot 0 = 0 \tag{67}$$

This holds for all active neurons. For dead neurons ($\mathcal{F}_i = \emptyset$), the gradient is automatically zero.

Step 5: (W_D^*, W_E^*) is a partial optimum.

By Theorem 3.2, $\tilde{\mathcal{L}}_{\text{SDL}}$ is convex in W_D for fixed W_E^*, and convex in W_E for fixed W_D^* within $\Omega_{\mathcal{A}}$.

Since we have verified:

- $\nabla_{W_D} \tilde{\mathcal{L}}_{\text{SDL}}(W_D^*, W_E^*) = 0$ (Step 3)
- $\nabla_{W_E} \tilde{\mathcal{L}}_{\text{SDL}}(W_D^*, W_E^*) = 0$ within $\Omega_{\mathcal{A}}$ (Step 4)

By the first-order optimality condition for convex functions:

- W_D^* minimizes $\tilde{\mathcal{L}}_{\text{SDL}}(\cdot, W_E^*)$

- W_E^* minimizes $\tilde{\mathcal{L}}_{\text{SDL}}(W_D^*, \cdot)$ over $\Omega_{\mathcal{A}}$

By standard results in biconvex optimization (Gorski et al., 2007), (W_D^*, W_E^*) is a partial optimum.

Step 6: Non-zero loss.

Since the pattern is polysemantic, there exists $i \in [n_q]$ with $|\mathcal{F}_i| \geq 2$. Choose distinct $d_1, d_2 \in \mathcal{F}_i$.

By Theorem 3.4, zero loss requires:

$$w_r^{d_1} = W_D^* z_{d_1} = W_D^{*,i}(z_{d_1})_i, \quad w_r^{d_2} = W_D^* z_{d_2} = W_D^{*,i}(z_{d_2})_i \tag{68}$$

If both equations held, then:

$$\frac{w_r^{d_1}}{(z_{d_1})_i} = W_D^{*,i} = \frac{w_r^{d_2}}{(z_{d_2})_i} \tag{69}$$

Since $(z_{d_1})_i = \langle w_E^{*,i}, w_p^{d_1} \rangle > 0$ and $(z_{d_2})_i = \langle w_E^{*,i}, w_p^{d_2} \rangle > 0$, this requires:

$$w_r^{d_1} = \frac{(z_{d_1})_i}{(z_{d_2})_i} w_r^{d_2} \tag{70}$$

This means $w_r^{d_1} \propto w_r^{d_2}$, contradicting Assumption 2.1 (distinct features have linearly independent reconstruction vectors). Therefore:

$$\tilde{\mathcal{L}}_{\text{SDL}}(W_D^*, W_E^*) = \sum_{d=1}^{n} M_d \|w_r^d - W_D^* z_d\|^2 > 0 \tag{71}$$

Conclusion. We have constructed (W_D^*, W_E^*) exhibiting the polysemanticity pattern, explicitly verified both gradients vanish, and shown it achieves positive loss, completing the proof. □

J.6 Proof of Theorem 3.8

Proof. We construct a realizable pattern exhibiting feature absorption by scaling down the parent neuron's encoder until one sub-concept separates, then adding a dedicated neuron for the separated feature.

Step 1: Initial configuration and feature geometry.

Since $\mathcal{P} = (\mathcal{F}_1, \ldots, \mathcal{F}_M)$ is realizable, there exists an encoder $W_E \in \mathbb{R}^{M \times n_p}$ and threshold $c > 0$ such that:

$$\forall i \in [M], \quad \mathcal{F}_i = \{d \in [n] : \langle w_E^i, w_p^d \rangle > c\} \tag{72}$$

Fix any parent neuron $i^* \in [M]$ with sub-concepts $\mathcal{F}_{i^*} = \{d_{i^*,1}, \ldots, d_{i^*,k_{i^*}}\}$ where $k_{i^*} \geq 2$.

For each sub-concept, compute its activation strength:

$$a_j := \langle w_E^{i^*}, w_p^{d_{i^*,j}} \rangle > c, \quad \forall j \in [k_{i^*}] \tag{73}$$

Since there are finitely many sub-concepts, define:

$$a_{\min} := \min_{j \in [k_{i^*}]} a_j > c \tag{74}$$

Let $j^* \in [k_{i^*}]$ be the index achieving this minimum: $a_{j^*} = a_{\min}$.

Step 2: Scale down parent neuron to separate one sub-concept.

We shrink the encoder row $w_E^{i^*}$ by multiplying with scaling factor $\lambda \in (0, 1)$:

$$\tilde{w}_E^{i^*} := \lambda \cdot w_E^{i^*} \tag{75}$$

The new activation strengths become:

$$\tilde{a}_j := \langle \tilde{w}_E^{i^*}, w_p^{d_{i^*,j}} \rangle = \lambda \cdot a_j \tag{76}$$

Choose λ such that:

$$\lambda := \frac{c + a_{\min}}{2a_{\min}} \in \left(\frac{c}{a_{\min}}, 1 \right) \tag{77}$$

This gives:

- For the minimum-activation feature: $\tilde{a}_{j^*} = \lambda a_{\min} = \frac{c+a_{\min}}{2} < c + \epsilon$ for any small $\epsilon > 0$
- For all other features: $\tilde{a}_j = \lambda a_j > \lambda a_{\min} = \frac{c+a_{\min}}{2}$

By choosing the threshold to be $c' = \frac{c+a_{\min}}{2}$, we have:

$$\tilde{a}_{j^*} = c' \quad \text{(at boundary)}, \quad \tilde{a}_j > c' \text{ for all } j \neq j^* \tag{78}$$

Slightly increasing c' by infinitesimal $\epsilon > 0$ gives $c'' = c' + \epsilon$, ensuring:

$$\tilde{a}_{j^*} < c'' < \tilde{a}_j \quad \forall j \neq j^* \tag{79}$$

Therefore, with encoder row $\tilde{w}_E^{i^*}$ and threshold c'':

$$\{d \in [n] : \langle \tilde{w}_E^{i^*}, w_p^d \rangle > c''\} = \mathcal{F}_{i^*} \setminus \{d_{i^*,j^*}\} \tag{80}$$

Step 3: Construct dedicated neuron for separated feature.

We create a new encoder row parallel to the separated feature:

$$w_E^{new} := \alpha \cdot w_p^{d_{i^*,j^*}} \tag{81}$$

where $\alpha > 0$ is chosen sufficiently large.

By the unit-norm assumption ($\|w_p^{d_{i^*,j^*}}\| = 1$):

$$\langle w_E^{new}, w_p^{d_{i^*,j^*}} \rangle = \alpha \|w_p^{d_{i^*,j^*}}\|^2 = \alpha \tag{82}$$

For any other feature $d \neq d_{i^*,j^*}$:

$$\langle w_E^{new}, w_p^d \rangle = \alpha \langle w_p^{d_{i^*,j^*}}, w_p^d \rangle \leq \alpha M \tag{83}$$

where $M < 1$ is the maximum interference.

Choose α large enough such that:

$$\alpha > c'', \quad \alpha M < c'' \tag{84}$$

This is possible by taking $\alpha = \frac{2c''}{1+M}$, giving:

$$\alpha = \frac{2c''}{1+M} > c'', \quad \alpha M = \frac{2c''M}{1+M} < c'' \tag{85}$$

where the first inequality uses $M < 1$ and the second uses $M < 1$.

Therefore:

- $\langle w_E^{new}, w_p^{d_{i^*,j^*}} \rangle = \alpha > c''$ (separated feature activates new neuron)

- $\langle w_E^{new}, w_p^d \rangle \leq \alpha M < c''$ for all $d \neq d_{i^*,j^*}$ (no other features activate)

Step 4: The new activation pattern.

Construct the expanded encoder $W'_E \in \mathbb{R}^{(M+1)\times n_p}$:

$$W'_E = \begin{bmatrix} w_E^1 \\ \vdots \\ w_E^{i^*-1} \\ \tilde{w}_E^{i^*} \\ w_E^{i^*+1} \\ \vdots \\ w_E^M \\ w_E^{new} \end{bmatrix} \tag{86}$$

With threshold c'', the resulting activation pattern is:

$$\mathcal{P}' = (\mathcal{F}_1, \ldots, \mathcal{F}_{i^*-1}, \mathcal{F}_{i^*} \setminus \{d_{i^*,j^*}\}, \mathcal{F}_{i^*+1}, \ldots, \mathcal{F}_M, \{d_{i^*,j^*}\}) \tag{87}$$

This pattern is realizable by construction. □