A UNIFIED THEORETICAL FRAMEWORK OF SPARSE DICTIONARY LEARNING IN MECHANISTIC INTERPRETABILITY

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 formulates major SDL variants within a common optimization problem and show that, under sparse linear representation assumptions, its feature-wise approximation exhibits piecewise biconvex structure. We characterize the zero-loss conditions and spurious partial minima of this objective, exposing the underdetermined nature of reconstruction-based SDL and providing principled explanations for feature absorption and dead neurons. Furthermore, we demonstrate how to reduce this underdetermination with a novel technique, feature anchoring, as an extension of our theory.

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 sub-concepts 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

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

mere implementation artifacts but reflect fundamental structural properties of the SDL optimization problem itself, warranting rigorous theoretical analysis of their root causes.

We argue that a fundamental source of these failure modes is the non-identifiability of SDL methods: even under the idealized Linear Representation Hypothesis, SDL optimization can admit multiple solutions achieving the same reconstruction objective without uniquely recovering the underlying interpretable features, necessitating comprehensive theoretical analysis of 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, including 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. Under sparse linear representation assumptions, we approximate the SDL population objective by a feature-wise reconstruction objective and characterize its piecewise biconvex structure. We then analyze its global solution set and zero-loss conditions, exposing the underdetermined nature of reconstruction-based SDL: low or even zero reconstruction loss does not in general uniquely determine the underlying feature dictionary. We further establish the prevalence of spurious partial minima exhibiting polysemanticity and connect these solutions to feature absorption (Chanin et al., 2025) and dead neurons. Furthermore, we demonstrate how to reduce the underdetermined nature of SDL methods with a novel technique, feature anchoring, as an extension of our theory, proving the possibility to utilize our theory for algorithmic improvements, with details described in the Appendix. Our main contributions are as follows:

- We build the first unified theoretical framework for SDL in mechanistic interpretability as a general optimization problem encompassing diverse SDL methods.
- We theoretically characterize the piecewise biconvex structure of SDL optimization under sparse linear representation assumptions, identifying the corresponding mathematical object for SDL methods in mechanistic interpretability.
- We characterize the SDL optimization landscape and its non-identifiability, and establish the prevalence of spurious partial minima, providing principled explanations for phenomena including polysemanticity, feature absorption, and dead neurons.

2 Related Works

2.1 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.

2.2 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).

2.3 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.

3 Preliminaries

In this section, we present a unified theoretical framework for Sparse Dictionary Learning (SDL). We begin with formal definitions of foundational concepts, then introduce our general optimization framework and show how various SDL methods instantiate it. The representation assumptions used for our theoretical analysis are deferred to Appendix F.

3.1 Input Distribution and Model Representation

Definition 3.1 (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 3.2 (Model Representation)**.** *For each $s \sim \mathcal{D}$, the network produces an input representation $\mathbf{x}_p(s) \in \mathbb{R}^{n_p}$ and a target representation $\mathbf{x}_r(s) \in \mathbb{R}^{n_r}$, which are directly observable by running the model on s (we provide a full notation list in Appendix E).*

3.2 General Optimization Framework for SDL

We formalize SDL as a general optimization problem over an encoder and decoder. Following previous theoretical analyses (Cui et al., 2025; Elhage et al., 2022), we omit bias terms for mathematical simplicity.

Definition 3.3 (Sparse Dictionary Learning)**.** *An 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 squared error:

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

For the theoretical analysis, under Assumption F.1 in Appendix F, 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:

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

3.3 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 D).

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).

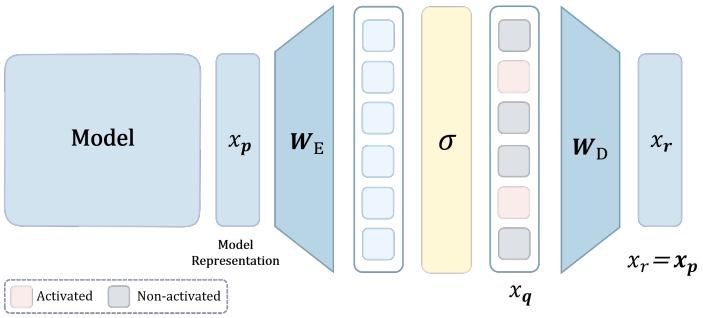

Figure 1: Sparse Autoencoder: encoder W_E maps $\mathbf{x}_p$ to sparse latents $\mathbf{x}_q$, decoder W_D reconstructs from $\mathbf{x}_q$.

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 an 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).

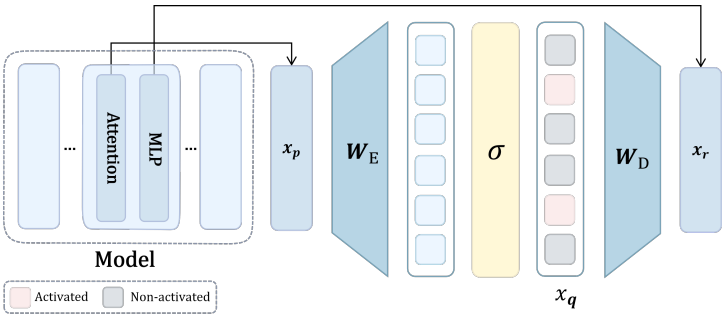

Figure 2: Transcoder: encoder W_E maps $\mathbf{x}_p(s)$ to sparse latents $\mathbf{x}_q(s)$, decoder W_D gives $\mathbf{x}_r(s)$ as a prediction of MLP's output.

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).

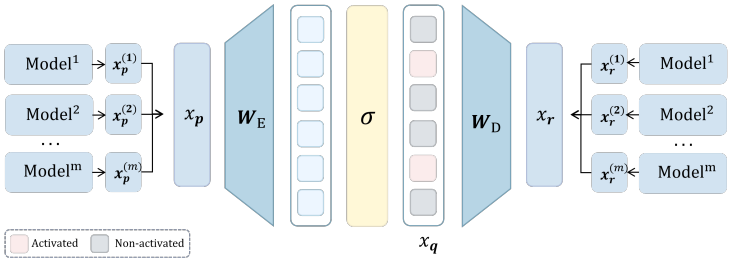

Figure 3: Crosscoder: encoder W_E maps concatenated multi-layer input $\mathbf{x}_p$ to $\mathbf{x}_q$, decoder W_D reconstructs multi-layer output $\mathbf{x}_r$.

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 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 3.3) 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.

4 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 4.1 establishes a loss approximation enabling tractable analysis (Theorem 4.1). Section 4.2 proves SDL exhibits piecewise biconvex structure within activation pattern regions (Theorem 4.2). Section 4.3 characterizes feature-recovering solutions and the zero-loss solution set, revealing the underdetermined nature of reconstruction-based SDL (Theorems 4.3 and 4.4). Section 4.4 establishes the existence of broad families of spurious partial minima exhibiting polysemanticity (Theorem 4.6). Section 4.5 connects hierarchical concept structures and feature-absorption patterns to these partial minima (Theorem 4.8). We provide full proofs along with the assumptions used in our analysis in Appendix F.

4.1 Approximate Loss with Feature Reconstruction

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

Theorem 4.1 (Loss Approximation)**.** *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.

4.2 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 4.2 (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 4.1), define the corresponding activation pattern region as:*

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

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 and enabling theoretical analysis of the optimization landscape.

4.3 Characterizing Feature-Recovering and Zero-Loss Solutions

Characterizing feature-recovering and zero-loss solutions reveals both the success conditions for SDL training and the fundamental underdetermined nature of the optimization problem.

Theorem 4.3 (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 4.4 (Necessary and Sufficient Conditions for Zero Loss)**.** *Assume $M_d > 0$ for all $d \in [n]$. The approximate loss satisfies $\tilde{\mathcal{L}}_{SDL}(W_D, W_E) = 0$ if and only if*

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

Theorem 4.3 provides a constructive feature-aligned solution with bounded reconstruction loss, while Theorem 4.4 characterizes the zero-loss solution set. These reconstruction constraints need not uniquely determine the encoder and decoder, allowing distinct parameterizations beyond the feature-recovering configuration. Critically, SDL can achieve near-zero reconstruction loss without recovering the ground-truth features, as illustrated empirically in Figure 4.

4.4 Characterizing Spurious Partial Minima of SDL

Beyond feature-recovering solutions, SDL optimization exhibits spurious partial minima where neurons exhibit polysemanticity—responding to multiple unrelated features.

Example 4.5 (Spurious Partial Minimum)**.** *Consider $n = n_p = n_q = n_r = 2$, $\sigma = \sigma_{Top\text{-}1} \circ \sigma_{ReLU}$, $S \to 1$, and $M_1 = M_2 = 1$. Let:*

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

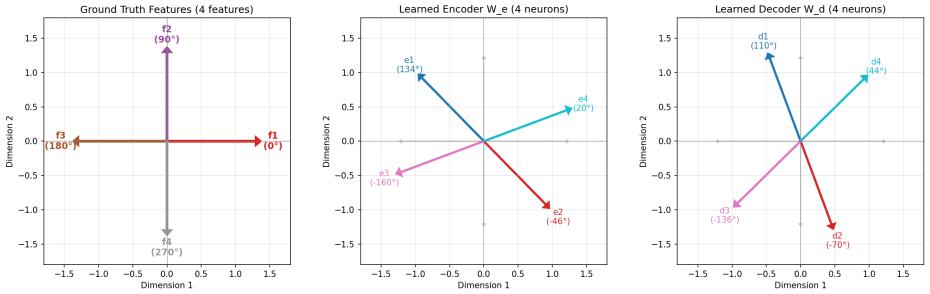

Figure 4: **Near-zero reconstruction loss without recovering ground-truth features.** We design the Linear Representation Bench to provide full knowledge of the ground-truth features for studying SDL methods. We observe that these methods can achieve near-zero loss without recovering the ground-truth 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, illustrating the underdetermined nature of reconstruction-based SDL optimization.

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 corresponding fixed activation pattern, 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 4.1 (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 4.6 (Spurious Partial Minima)**.** *For any activation pattern* $\mathcal{P} = (\mathcal{F}_1, \ldots, \mathcal{F}_{n_q})$ *that is realizable, polysemantic, and forms a partition of* $[n]$*, and for which the corresponding within-pattern optimization admits a blockwise minimizer with nonzero reconstruction residual, there exists a partial minimum* (W_D^*, W_E^*) *of* $\tilde{\mathcal{L}}_{SDL}$ *exhibiting this pattern with positive loss.*

This establishes a broad family of spurious partial minima in SDL optimization: realizable polysemantic activation patterns satisfying the conditions of Theorem 4.6 can correspond to suboptimal partial optima. This characterizes a structural source of persistent polysemantic solutions without making assumptions about whether gradient-based training converges to them.

4.5 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 connect feature absorption to the partial-minimum framework by considering hierarchical concept structures whose corresponding split activation patterns are realizable.

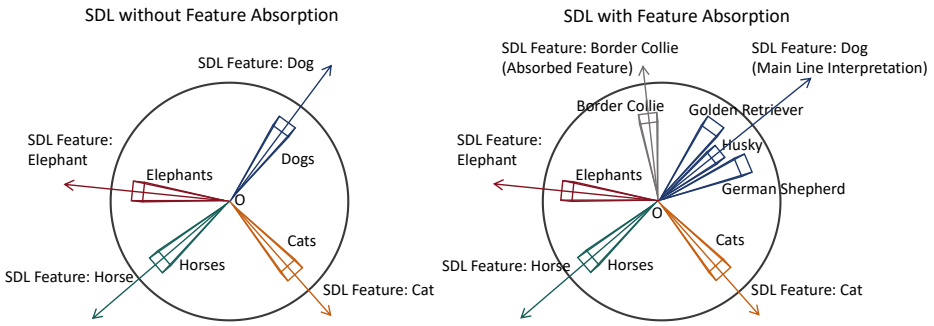

Figure 5: **Feature absorption and hierarchical concept structure.** Left: Ideal SDL features without absorption. Right: a hierarchical concept structure exhibiting feature absorption, where only a subset of the sub-concepts of "Dog" activates the corresponding SDL feature.

Example 4.7 (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 4.2 (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 4.8 (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 $k_i \geq 2$.*

For any $i^* \in [M]$ and $j^* \in [k_{i^*}]$, if the split activation pattern

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

is realizable and satisfies the conditions of Theorem 4.6, then $\mathcal{P}'$ corresponds to a spurious partial minimum exhibiting feature absorption.

Theorem 4.8 connects feature absorption to the partial-minimum structure of SDL optimization: whenever a hierarchical split activation pattern is realizable and satisfies the conditions of Theorem 4.6, it can manifest as a spurious partial minimum, providing a theoretical mechanism for this widely observed empirical phenomenon.

5 Conclusion

We develop a unified theoretical framework for Sparse Dictionary Learning in mechanistic interpretability, showing that a broad family of SDL methods can be understood through a common optimization formulation. Within this framework, we establish the piecewise biconvex structure of SDL optimization, characterize feature-recovering and zero-loss solutions, and expose the underdetermined nature of reconstruction-based SDL: low or even zero reconstruction loss does not in general uniquely determine the underlying interpretable features. We further establish broad families of spurious partial minima exhibiting polysemanticity and connect hierarchical concept structures and feature absorption to these partial minima. Together, these results provide a principled account of several persistent empirical pathologies observed in modern SDL methods and clarify the structural limitations of reconstruction-based objectives. We additionally provide controlled empirical analyses and a theory-guided extension in the appendix to illustrate how these theoretical insights can inform the design and evaluation of future SDL methods.

AI USE STATEMENT

In this work, we used generative AI tools, including ChatGPT, Claude, and Claude Code to implement methods, write and edit code, and assist in drafting the manuscript. We have reviewed all AI-assisted work: code was checked against expected behavior and the underlying data, and all AI-drafted text and result interpretations were verified against experimental outputs and revised where necessary. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

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.

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 Discussion

A.1 What Does the Biconvex Structure Tell Us?

The piecewise biconvex structure of SDL optimization (Theorem 4.2) provides a common mathematical view of an otherwise highly non-convex problem. Within each fixed activation region, the approximate SDL objective is convex in the encoder and decoder separately, allowing its optimization landscape to be analyzed through the theory of biconvex optimization. This structure underlies our subsequent characterization of underdetermination and spurious partial minima, and connects modern SDL methods in mechanistic interpretability to a broader body of classical optimization theory. Importantly, piecewise biconvexity does not imply that SDL is easy to optimize globally: transitions between activation regions remain non-convex, and even within a region, biconvex objectives can admit suboptimal partial minima.

A.2 Reconstruction Does Not Imply Feature Recovery

Theorem 4.4 highlights a fundamental distinction between reconstruction and identification. SDL is typically trained by minimizing reconstruction error, yet this objective constrains the composition of the encoder and decoder without in general uniquely determining their individual feature directions. Consequently, distinct parameterizations can implement the same reconstruction behavior, and empirically we observe solutions that achieve near-perfect reconstruction without recovering the underlying ground-truth features. This suggests that reconstruction quality alone is insufficient evidence that learned SDL features correspond to the concepts encoded by the original representation. More broadly, evaluation of SDL methods should distinguish reconstruction performance from feature recovery and interpretability.

A.3 Spurious Partial Minima as a Source of SDL Pathologies

Theorems 4.6 and 4.8 provide an optimization perspective on polysemanticity and feature absorption. Realizable polysemantic activation patterns satisfying the conditions of Theorem 4.6 can correspond to suboptimal partial minima. Feature absorption can arise as a structured instance of this phenomenon when hierarchical concepts admit corresponding split activation patterns: one subconcept separates while related sub-concepts remain represented collectively. These results show that commonly observed SDL pathologies can arise from the geometry of the optimization landscape itself, rather than solely from insufficient model capacity or imperfect training procedures. Our analysis concerns the existence of these partial minima and does not imply that gradient-based optimization necessarily converges to them.

A.4 On Theoretical Assumptions

Our analysis deliberately studies an idealized regime in which representations admit a sparse linear feature decomposition. Different results additionally rely on particular conditions, including feature independence, high sparsity, and masking-based activation operators. These assumptions make the optimization landscape analytically tractable and allow us to isolate structural properties of SDL objectives. They should not be interpreted as exact descriptions of representations in real neural networks. Rather, our results characterize structural ambiguities and spurious partial minima that can already arise in this simplified setting. Understanding how these phenomena change as the assumptions are progressively relaxed is an important direction for future theoretical work.

B Limitations

Our theoretical framework establishes structural properties of SDL optimization under an idealized representation model, but several limitations remain:

- **Linear Feature Representation.** Our theoretical analysis assumes that model representations admit an exact linear decomposition into underlying feature directions. Real neural network representations may contain features with more complex geometry that cannot be captured by a single linear direction (Engels et al., 2025).

- **High-Sparsity Regime.** The feature-wise loss approximation assumes a high-sparsity regime, $(1 - S) \leq 1/n$. As feature co-activation becomes more frequent, the approximation error increases and interactions between simultaneously active features must be modeled explicitly.
- **Feature Independence.** Our approximation assumes mutually independent ground-truth feature activations. Real-world concepts can exhibit substantial statistical dependence, and such correlations may introduce additional structure into the optimization landscape.
- **Restricted Activation Structure.** Our biconvex analysis applies to sparsity mechanisms that act as masking operators within fixed activation regions, including ReLU, JumpReLU, Top-k, and related variants. SDL architectures with substantially different nonlinearities may not exhibit the same piecewise biconvex structure.
- **Partial Minima versus Optimization Dynamics.** We characterize classes of spurious partial minima but do not establish that gradient-based training necessarily converges to these solutions, nor do we characterize their basins of attraction. Connecting the static landscape characterized here to actual training dynamics remains an open problem.
- **Population-Level Analysis.** Our theoretical results are formulated in terms of the population objective under the input distribution $\mathcal{D}$. Finite-sample effects may introduce additional sources of ambiguity and optimization error that are not captured by the present analysis.

C Future Work

Our results suggest several directions for extending the theoretical understanding of SDL:

- **Beyond the High-Sparsity Regime.** Extending the analysis to moderate sparsity would require explicitly accounting for feature co-activation and interference terms rather than relying on feature-wise reconstruction. Such results would clarify whether the piecewise biconvex structure and associated landscape phenomena persist in more realistic regimes.
- **Correlated and Nonlinear Features.** Relaxing feature independence and exact linearity would broaden the framework to representations with correlated concepts, approximately linear features, and more general feature geometries.
- **Optimization Dynamics and Convergence.** Our characterization of spurious partial minima motivates studying when gradient-based optimization enters or escapes their basins of attraction. Establishing convergence guarantees and identifying the conditions governing convergence to different classes of solutions would connect our landscape results more directly to practical SDL training.
- **Identifiability Conditions.** The underdetermined nature of reconstruction-based SDL raises the complementary question of what additional assumptions or constraints are sufficient to identify the underlying feature dictionary. Characterizing such conditions could provide theoretical guidance for the design of future SDL objectives.
- **Algorithms Exploiting Biconvex Structure.** The piecewise biconvex characterization motivates investigating optimization procedures designed specifically for biconvex problems, including alternating convex search and global optimization approaches (Gorski et al., 2007). An important question is whether these methods can improve optimization within activation regions or provide stronger guarantees than standard gradient-based training.
- **Finite-Sample Theory.** Extending the population-level results to empirical objectives would enable analysis of sample complexity, generalization of learned dictionaries, and the interaction between finite-data effects and the intrinsic underdetermination of SDL.

D TAXONOMY FOR SDL IN MECHANISTIC INTERPRETABILITY

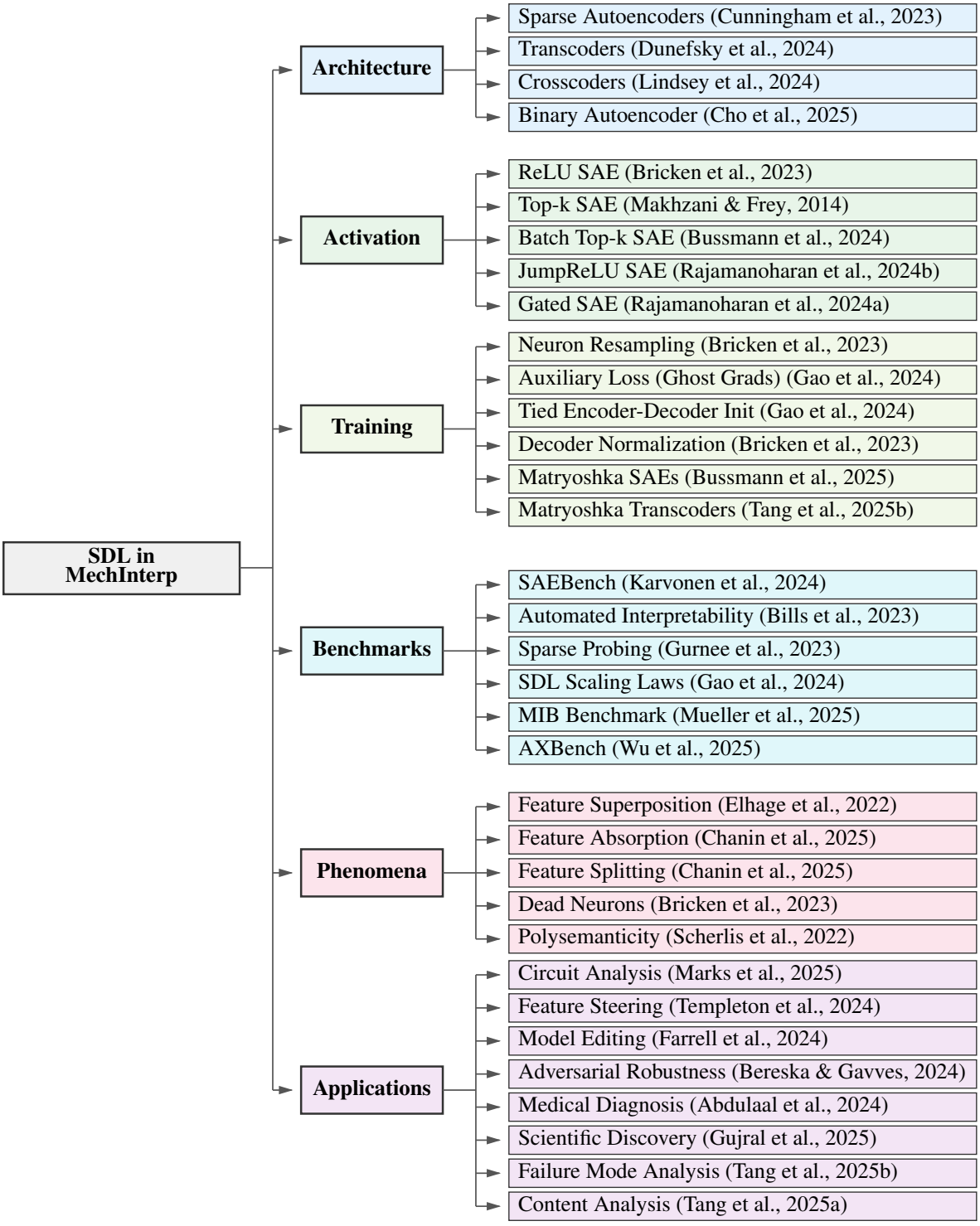

Figure 6: Hierarchical taxonomy of SDL research in Mechanistic Interpretability.

E 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 feature activations 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	Ground-truth feature matrix for $\mathbf{x}_p$ ($n_p \times n$)
W_r	Ground-truth feature matrix for $\mathbf{x}_r$ ($n_r \times n$)
$\mathbf{w}_p^d$	d-th column of W_p (input feature direction for feature d)
$\mathbf{w}_r^d$	d-th column of W_r (target feature direction for feature d)
W_p^{true}	Ground-truth input feature matrix (Linear Representation Bench)
W_r^{true}	Ground-truth target feature matrix (Linear Representation Bench)
W_p^{learned}	Learned input feature matrix ($W_E^\top$)
Sparsity and Interference	
S	Sparsity level: $\Pr(x_d(s) = 0) \geq S$ for all $d \in [n]$
M	Maximum interference: $\max_{i \neq j} \langle \mathbf{w}_p^i, \mathbf{w}_p^j \rangle$
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 feature-wise SDL loss
$\mathcal{L}_{\text{anchor}}$	Feature anchoring penalty
$\mathcal{L}_{\text{SDL-FA}}$	Anchored SDL objective: $\mathcal{L}_{\text{SDL}} + \lambda_{\text{anchor}}\mathcal{L}_{\text{anchor}}$
Activation Patterns	
$\mathcal{P} = (\mathcal{F}_1, \ldots, \mathcal{F}_{n_q})$	Activation pattern
$\mathcal{F}_i$	Set of ground-truth features activating neuron i
$\mathcal{A}(d)$	Set of neurons activated by feature d
$\Omega_{\mathcal{P}}$	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

F PROOFS OF THE THEOREMS

F.1 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 satisfying the following assumptions.

Assumption F.1 (Representation Assumptions)**.** *There exists a* feature function $\mathbf{x} : \mathcal{X} \to \mathbb{R}^n$ *and feature matrices* $W_p \in \mathbb{R}^{n_p \times n}$ *and* $W_r \in \mathbb{R}^{n_r \times n}$ *such that:*

1. ***Linear Decomposition****: For all* $s \sim \mathcal{D}$*,*
$$\mathbf{x}_p(s) = W_p \mathbf{x}(s), \qquad \mathbf{x}_r(s) = W_r \mathbf{x}(s).$$
2. ***Unit Norm****: Both feature matrices have unit-norm columns:*
$$\|W_p[:, i]\|_2 = \|W_r[:, i]\|_2 = 1 \quad \forall i \in [n].$$
3. ***Non-negativity****:* $\mathbf{x}(s) \in \mathbb{R}^n_+$.
4. ***Sparsity and Non-degeneracy****: There exists* $S \in [0, 1]$ *such that* $\forall i \in [n]$*,*
$$\Pr_{s\sim\mathcal{D}}(x_i(s) = 0) \geq S, \qquad \Pr_{s\sim\mathcal{D}}(x_i(s) > 0) > 0.$$
5. ***Independence****: The component activations* $\{x_i(s)\}_{i=1}^n$ *are mutually independent.*

We refer to $\mathbf{x}(s)$ *as the* ground-truth feature activations*, 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).*

For Theorem 4.1, we additionally assume that the activation function is positively homogeneous on non-negative scalings:

$$\sigma(\alpha z) = \alpha \sigma(z) \qquad \forall \alpha \geq 0. \tag{22}$$

This condition allows the magnitude of an isolated ground-truth feature to factor out of the SDL reconstruction. It is satisfied by ReLU and Top-k masking under positive rescaling; activation mechanisms with fixed nonzero thresholds require a magnitude-dependent version of the feature-wise approximation.

F.2 PROOF OF THEOREM 4.1

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:

$$\mathcal{L}_{\text{SDL}} = \sum_{m=0}^{n} \Pr(\|\mathbf{x}(s)\|_0 = m) \cdot \mathbb{E}\big[\|\mathbf{x}_r(s) - W_D\sigma(W_E\mathbf{x}_p(s))\|^2 \mid \|\mathbf{x}(s)\|_0 = m\big]. \tag{23}$$

The $m = 0$ term contributes zero since $\sigma(0) = 0$. For $m = 1$, exactly one feature d is active, giving $\mathbf{x}_r(s) = x_d(s)w_r^d$ and $\mathbf{x}_p(s) = x_d(s)w_p^d$. By positive homogeneity,

$$W_D\sigma(W_E(x_d(s)w_p^d)) = x_d(s)W_D\sigma(W_E w_p^d). \tag{24}$$

Hence the contribution from singleton activations is

$$\sum_{d=1}^{n} M_d \left\|w_r^d - W_D\sigma(W_E w_p^d)\right\|^2 = \tilde{\mathcal{L}}_{\text{SDL}}(W_D, W_E), \tag{25}$$

where

$$M_d := \Pr(\mathbf{x}(s) = x_d(s)\mathbf{e}_d) \cdot \mathbb{E}[x_d(s)^2 \mid x_d(s) > 0]. \tag{26}$$

For $m \geq 2$, each feature is active with probability at most $(1 - S)$. By independence and a union bound over pairs,

$$\Pr(\|\mathbf{x}(s)\|_0 \geq 2) \leq \sum_{i<j} \Pr(x_i(s) > 0, x_j(s) > 0) \leq \binom{n}{2}(1-S)^2 \leq n^2(1-S)^2. \tag{27}$$

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

$$\left|\mathcal{L}_{\text{SDL}} - \tilde{\mathcal{L}}_{\text{SDL}}\right| \leq n^2 C(1-S)^2. \tag{28}$$

□

F.3 Proof of Theorem 4.2

Proof. **Preliminary observation.** Within a fixed masking region $\Omega_{\mathcal{P}}$, the activation mask is constant. Therefore, for any $W_E \in \Omega_{\mathcal{P}}$, $z_d(W_E) := \sigma(W_E w_p^d)$ satisfies

$$z_d(W_E)_i = \begin{cases} \langle w_E^i, w_p^d \rangle, & \text{if neuron } i \text{ is active for feature } d, \\ 0, & \text{otherwise.} \end{cases} \tag{29}$$

Hence $z_d(W_E)$ is linear in W_E within $\Omega_{\mathcal{P}}$.

Per-feature decomposition. Define

$$f_d(W_D, W_E) := \|w_r^d - W_D z_d(W_E)\|^2. \tag{30}$$

Then

$$\tilde{\mathcal{L}}_{\text{SDL}}(W_D, W_E) = \sum_{d=1}^{n} M_d f_d(W_D, W_E), \qquad M_d > 0. \tag{31}$$

Convexity in W_D. For fixed $W_E \in \Omega_{\mathcal{P}}$, z_d is fixed, so

$$f_d(W_D) = \|w_r^d - W_D z_d\|^2 \tag{32}$$

is a squared affine function of W_D and is therefore convex.

Convexity in W_E. For fixed W_D, the fixed activation mask gives

$$W_D z_d(W_E) = \sum_{i \in \mathcal{A}(d)} \langle w_E^i, w_p^d \rangle w_D^i, \tag{33}$$

which is affine in W_E. Hence

$$f_d(W_E) = \|w_r^d - W_D z_d(W_E)\|^2 \tag{34}$$

is convex in W_E within every convex fixed-mask activation region $\Omega_{\mathcal{P}}$.

Since non-negative linear combinations preserve convexity in each block, $\tilde{\mathcal{L}}_{\text{SDL}}$ is biconvex over $\mathbb{R}^{n_r \times n_q} \times \Omega_{\mathcal{P}}$ for every convex fixed-mask activation region. □

F.4 Proof of Theorem 4.3

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

$$(W_E^* w_p^d)_k = \langle w_p^k, w_p^d \rangle. \tag{35}$$

By the unit-norm condition in Assumption F.1,

$$(W_E^* w_p^d)_d = 1. \tag{36}$$

Define

$$\mathcal{A}_d := \{k \in [n] : (\sigma(W_E^* w_p^d))_k > 0\}. \tag{37}$$

For every activated $k \neq d$, the masking property gives

$$0 < \langle w_p^k, w_p^d \rangle \leq M. \tag{38}$$

The reconstruction is therefore

$$W_D^* \sigma(W_E^* w_p^d) = \sum_{k \in \mathcal{A}_d} \langle w_p^k, w_p^d \rangle w_r^k, \tag{39}$$

and the reconstruction error satisfies

$$w_r^d - W_D^* \sigma(W_E^* w_p^d) = - \sum_{k \in \mathcal{A}_d \setminus \{d\}} \langle w_p^k, w_p^d \rangle w_r^k. \tag{40}$$

Using the triangle inequality, $\|w_r^k\|_2 = 1$, and $\langle w_p^k, w_p^d \rangle \leq M$ for activated $k \neq d$,

$$\left\| w_r^d - W_D^* \sigma(W_E^* w_p^d) \right\|_2 \leq M(K_d - 1), \tag{41}$$

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

$$\tilde{\mathcal{L}}_{\text{SDL}}(W_D^*, W_E^*) \leq M^2 \sum_{d=1}^{n} M_d (K_d - 1)^2 \tag{42}$$

$$\leq n^2 M^2 \sum_{d=1}^{n} M_d. \tag{43}$$

□

F.5 Proof of Theorem 4.4

Proof. We prove both directions of the equivalence.

Sufficient condition. Suppose

$$w_r^d = W_D \sigma(W_E w_p^d) \qquad \forall d \in [n]. \tag{44}$$

Then

$$\tilde{\mathcal{L}}_{\text{SDL}}(W_D, W_E) = \sum_{d=1}^{n} M_d \|w_r^d - W_D \sigma(W_E w_p^d)\|^2 \tag{45}$$

$$= 0. \tag{46}$$

Necessary condition. Suppose $\tilde{\mathcal{L}}_{\text{SDL}}(W_D, W_E) = 0$. By Assumption F.1, independence and non-degeneracy imply

$$\Pr(\mathbf{x}(s) = x_d(s)\mathbf{e}_d) > 0, \qquad \mathbb{E}[x_d(s)^2 \mid x_d(s) > 0] > 0, \tag{47}$$

and therefore $M_d > 0$ for every $d \in [n]$. Since every summand in

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

is non-negative, zero total loss implies

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

Therefore

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

□

F.6 Proof of Theorem 4.6

Proof. Let $\mathcal{P}$ be a realizable polysemantic activation pattern forming a partition of $[n]$ and satisfying the conditions of Theorem 4.6. By realizability, the corresponding activation region $\Omega_{\mathcal{P}}$ is non-empty. By assumption, the optimization of $\tilde{\mathcal{L}}_{\text{SDL}}$ within this region admits a blockwise minimizer (W_D^*, W_E^*) with nonzero reconstruction residual. Thus,

$$W_D^* \in \arg\min_{W_D} \tilde{\mathcal{L}}_{\text{SDL}}(W_D, W_E^*) \tag{51}$$

and

$$W_E^* \in \arg\min_{W_E \in \Omega_{\mathcal{P}}} \tilde{\mathcal{L}}_{\text{SDL}}(W_D^*, W_E). \tag{52}$$

Therefore (W_D^*, W_E^*) is a partial optimum in the sense of biconvex optimization (Gorski et al., 2007). Because $W_E^* \in \Omega_{\mathcal{P}}$, it exhibits the activation pattern $\mathcal{P}$, which is polysemantic by assumption. Finally, the assumed nonzero reconstruction residual and $M_d > 0$ imply

$$\tilde{\mathcal{L}}_{\text{SDL}}(W_D^*, W_E^*) > 0. \tag{53}$$

Hence (W_D^*, W_E^*) is a spurious polysemantic partial optimum with positive loss. □

F.7 Proof of Theorem 4.8

Proof. Fix $i^* \in [M]$ and $j^* \in [k_{i^*}]$, and consider the split activation pattern

$$\mathcal{P}' = (\mathcal{F}_1, \ldots, \mathcal{F}_{i^*} \setminus \{d_{i^*,j^*}\}, \{d_{i^*,j^*}\}, \ldots, \mathcal{F}_M). \tag{54}$$

By assumption, $\mathcal{P}'$ is realizable and satisfies the conditions of Theorem 4.6. Therefore, Theorem 4.6 guarantees a partial optimum (W_D^*, W_E^*) exhibiting $\mathcal{P}'$ with positive reconstruction loss.

By construction of $\mathcal{P}'$, the feature d_{i^*,j^*} activates a dedicated neuron, while the remaining sub-concepts in

$$\mathcal{F}_{i^*} \setminus \{d_{i^*,j^*}\} \tag{55}$$

activate another neuron collectively. This is precisely the feature-absorption activation pattern described in Definition 4.2 and Example 4.7. Hence the corresponding partial optimum exhibits feature absorption. □

G Empirical Analysis of Neuron Resampling

Our theoretical analysis shows that realizable activation patterns containing inactive or polysemantic neurons can correspond to spurious partial minima of the SDL objective (Theorem **??**). This motivates examining whether perturbations to inactive neurons can improve optimization in practice. We therefore study neuron resampling (Bricken et al., 2023), a commonly used training intervention that periodically reinitializes dead neurons toward under-reconstructed directions.

We train SAEs on Llama 3.1 8B Instruct activations from layer 12 (dimension 4096), using a latent dimension of 131,072 for 30,000 training steps on FineWeb-Edu (Penedo et al., 2024), with 4096 token activations per step. We compare standard SAE training with training in which dead neurons are resampled after 5000 steps. As shown in Figure 7, neuron resampling changes the subsequent optimization trajectory and results in a lower final reconstruction loss. While this experiment does not directly identify the encountered optimization states as the partial minima characterized by our theory, it is consistent with the prediction that inactive-neuron configurations can create unfavorable regions of the SDL optimization landscape.

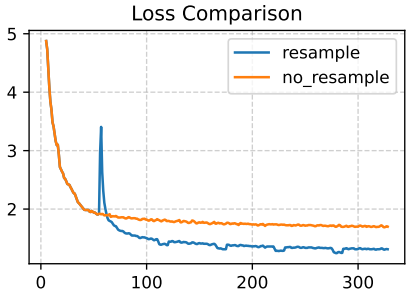

Figure 7: **Neuron resampling improves subsequent optimization.** Training curves on Llama 3.1 8B comparing standard SAE training with dead-neuron resampling introduced after 5000 steps (the x-axis is scaled by 100). Resampling changes the optimization trajectory and leads to a lower final reconstruction loss.

H Extension: Addressing the Underdetermined Nature of SDL Methods with Feature Anchoring

Our theoretical analysis reveals that SDL optimization can be fundamentally underdetermined: the zero-loss conditions in Theorem 4.4 constrain reconstruction without necessarily uniquely determining the underlying feature directions. In this section, we investigate this consequence empirically and explore whether introducing a small amount of semantic information can reduce the resulting ambiguity. We first introduce the **Linear Representation Bench**, a controlled synthetic setting that instantiates our representation assumptions while providing full access to the ground-truth features. We then introduce **feature anchoring**, a method-agnostic extension that constrains a subset of learned features to align with known semantic directions. We evaluate feature anchoring on both the Linear Representation Bench and real CLIP representations, provide qualitative examples of the resulting features, and systematically study its behavior across different degrees of superposition, interference, sparsity, and activation constraints.

H.1 The Linear Representation Bench

The Linear Representation Bench is a synthetic benchmark designed to instantiate the Representation Assumptions in Assumption F.1 with fully accessible ground-truth features. This controlled construction allows feature recovery to be measured directly, which is generally impossible for representations extracted from real neural networks. Figure 8 provides a low-dimensional visualization of the benchmark.

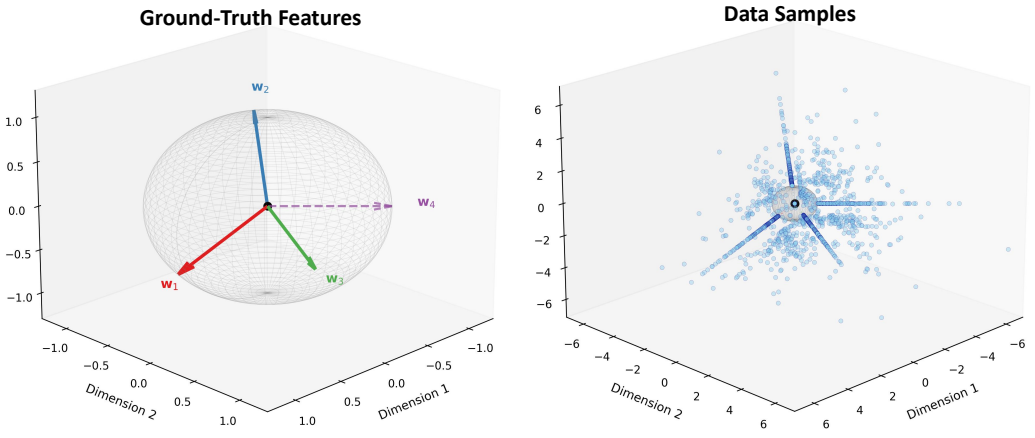

Figure 8: **Visualization of the Linear Representation Bench.** The example illustrates a three-dimensional representation space generated from four ground-truth feature directions. Observed representations are sparse non-negative combinations of these directions.

H.1.1 Ground-Truth Feature Matrix Generation

We construct the ground-truth input feature matrix $\mathbf{W}_p \in \mathbb{R}^{n_p \times n}$ to satisfy the representation assumptions while controlling the maximum interference between feature directions.

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

$$\mathbf{w}_p^d \leftarrow \frac{\mathbf{w}_p^d}{\|\mathbf{w}_p^d\|_2}, \qquad \forall d \in [n]. \tag{56}$$

Interference Minimization. We control pairwise interference using projected gradient descent. Define the soft-thresholded interference objective

$$\mathcal{L}_{\text{int}}(\mathbf{W}_p) = \sum_{i \neq j} \left[\max\left(0, \langle \mathbf{w}_p^i, \mathbf{w}_p^j \rangle - (M - \epsilon)\right)\right]^2 + \lambda \sum_{i \neq j} \left[\max\left(0, \langle \mathbf{w}_p^i, \mathbf{w}_p^j \rangle\right)\right]^2, \tag{57}$$

where M specifies the target maximum interference, $\epsilon > 0$ is a tolerance margin, and $\lambda > 0$ is a small regularization coefficient.

At each iteration, we perform a gradient update and project every feature direction back onto the unit sphere:

$$\mathbf{W}_p \leftarrow \mathbf{W}_p - \eta \nabla_{\mathbf{W}_p} \mathcal{L}_{\text{int}}, \qquad \mathbf{w}_p^d \leftarrow \frac{\mathbf{w}_p^d}{\|\mathbf{w}_p^d\|_2}, \qquad \forall d \in [n]. \tag{58}$$

Optimization continues until

$$\max_{i \neq j} \langle \mathbf{w}_p^i, \mathbf{w}_p^j \rangle \leq M. \tag{59}$$

For SDL variants with a target representation distinct from the input representation, we analogously construct a unit-normalized target feature matrix $\mathbf{W}_r \in \mathbb{R}^{n_r \times n}$. For self-reconstructing SAEs, we set $\mathbf{W}_r = \mathbf{W}_p$.

H.1.2 Sparse Ground-Truth Feature Generation

For each sample $i \in [N]$, we generate a sparse non-negative ground-truth feature vector $\mathbf{x}^{(i)} \in \mathbb{R}_+^n$.

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

$$m_d^{(i)} \sim \text{Bernoulli}(1 - \mathcal{S}). \tag{60}$$

Feature Magnitudes. For active features, magnitudes follow a shifted exponential distribution:

$$x_d^{(i)} = m_d^{(i)} \left(c_{\min} + \text{Exp}(\beta)\right), \tag{61}$$

where $c_{\min} \geq 0$ specifies a minimum activation magnitude and $\beta > 0$ controls the exponential distribution.

By construction, the generated features satisfy

$$x_d^{(i)} \geq 0, \qquad \Pr(x_d^{(i)} = 0) = \mathcal{S}, \tag{62}$$

and the components $\{x_d^{(i)}\}_{d=1}^n$ are mutually independent.

H.1.3 Representation Synthesis

Input representations are generated as sparse linear combinations of the ground-truth input directions:

$$\mathbf{x}_p^{(i)} = \mathbf{W}_p \mathbf{x}^{(i)} = \sum_{d=1}^n x_d^{(i)} \mathbf{w}_p^d. \tag{63}$$

For SDL variants with distinct target representations, we analogously generate

$$\mathbf{x}_r^{(i)} = \mathbf{W}_r \mathbf{x}^{(i)} = \sum_{d=1}^n x_d^{(i)} \mathbf{w}_r^d. \tag{64}$$

For SAEs, $\mathbf{W}_r = \mathbf{W}_p$ and therefore $\mathbf{x}_r^{(i)} = \mathbf{x}_p^{(i)}$. Thus, the resulting dataset satisfies Assumption F.1 by construction while retaining complete knowledge of the ground-truth feature activations and feature directions.

H.1.4 Default Configuration

Table 1 summarizes the default benchmark configuration used in our experiments.

Table 1: **Default configuration of 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

H.1.5 Feature Recovery Metrics

Given learned encoder directions $W_p^{\text{learned}} = W_E^\top \in \mathbb{R}^{n_p \times n_q}$ and ground-truth feature directions $W_p^{\text{true}} \in \mathbb{R}^{n_p \times n}$, we compute the similarity matrix

$$\mathbf{S} = \left| W_p^{\text{learned}\top} W_p^{\text{true}} \right| \in \mathbb{R}^{n_q \times n}. \tag{65}$$

For each ground-truth feature $i \in [n]$, we define its best-match similarity as

$$s_i = \max_j S_{ji}. \tag{66}$$

We use two complementary metrics. **GT Recovery** measures the fraction of evaluated ground-truth features whose best-match similarity exceeds a threshold τ, while **Maximum Inner Product** measures their average best-match similarity without thresholding. Let $\mathcal{E} \subseteq [n]$ denote the set of ground-truth features included in evaluation. For standard SDL training, $\mathcal{E} = [n]$; for feature anchoring, we exclude the anchor set $\mathcal{K}$ and use $\mathcal{E} = [n] \setminus \mathcal{K}$. The metrics are therefore

$$\mathbf{M}_{\text{GT}}(W_p^{\text{learned}}) = \frac{1}{|\mathcal{E}|} \sum_{i \in \mathcal{E}} \mathbf{1}\{s_i > \tau\}, \tag{67}$$

and

$$\mathbf{M}_{\text{IP}}(W_p^{\text{learned}}) = \frac{1}{|\mathcal{E}|} \sum_{i \in \mathcal{E}} s_i. \tag{68}$$

By excluding anchor features from evaluation, improvements under feature anchoring reflect recovery of previously unconstrained ground-truth directions rather than direct matching of the anchored features themselves.

H.2 Feature Anchoring

Theorem 4.4 shows that zero reconstruction constrains the encoder–decoder composition without, in general, uniquely specifying its constituent feature directions. Feature anchoring introduces partial semantic information into this optimization by constraining a subset of encoder and decoder directions to known reference features.

H.2.1 Anchor Feature Extraction

Feature anchoring assumes access to k anchor pairs $\{\tilde{\mathbf{w}}_p^{(i)}, \tilde{\mathbf{w}}_r^{(i)}\}_{i=1}^k$ corresponding to meaningful directions in the input and target representation spaces. We consider two mechanisms for constructing these anchors.

Ground-Truth Features. When ground-truth features are available, as in the Linear Representation Bench, we directly select a random subset $\mathcal{K} \subset [n]$ of k feature indices and use their corresponding directions:

$$\tilde{\mathbf{w}}_p^{(i)} = W_p^{\text{true}}[:, i], \qquad \tilde{\mathbf{w}}_r^{(i)} = W_r^{\text{true}}[:, i], \qquad i \in \mathcal{K}. \tag{69}$$

Subpopulation Mean Embeddings. For real representations, where ground-truth feature directions are unavailable, we use semantic subpopulations as proxies (Luo et al., 2024b). Given a labeled dataset $\mathcal{D} = \{(s_j, y_j)\}_{j=1}^{N}$, where $y_j \in \{1, \dots, C\}$, we compute the mean representation for each class:

$$\bar{\mathbf{x}}_p^{(c)} = \frac{1}{|\{j : y_j = c\}|} \sum_{j:y_j=c} \mathbf{x}_p(s_j), \tag{70}$$

and normalize it to obtain the corresponding semantic direction:

$$\tilde{\mathbf{w}}_p^{(c)} = \frac{\bar{\mathbf{x}}_p^{(c)}}{\|\bar{\mathbf{x}}_p^{(c)}\|_2}. \tag{71}$$

For self-reconstructing SDL methods such as SAEs, where $\mathbf{x}_r = \mathbf{x}_p$, the same semantic direction is used for both encoder and decoder anchors.

H.2.2 Anchored SDL Objective

Given k anchor pairs, we augment the standard SDL objective with an anchoring penalty that aligns the first k encoder rows and decoder columns with their corresponding semantic reference directions:

$$\mathcal{L}_{\text{SDL-FA}} = \mathcal{L}_{\text{SDL}} + \lambda_{\text{anchor}} \mathcal{L}_{\text{anchor}}, \tag{72}$$

where $\mathcal{L}_{\text{SDL}}$ denotes the standard SDL reconstruction objective in Equation 3, and

$$\begin{aligned} \mathcal{L}_{\text{anchor}} = &\left\| W_E[1{:}k, :] - [\tilde{\mathbf{w}}_p^{(1)}, \dots, \tilde{\mathbf{w}}_p^{(k)}]^\top \right\|_F^2 \\ &+ \left\| W_D[:, 1{:}k] - [\tilde{\mathbf{w}}_r^{(1)}, \dots, \tilde{\mathbf{w}}_r^{(k)}] \right\|_F^2. \end{aligned} \tag{73}$$

Feature anchoring therefore reduces degrees of freedom in the encoder–decoder factorization by encouraging a subset of learned directions to match semantic references. Because the penalty acts only on W_E and W_D, it can be applied to SAEs, transcoders, crosscoders, and their activation or loss variants.

H.3 Feature Anchoring on the Linear Representation Bench

We first test whether anchoring a subset of known ground-truth directions improves recovery of the remaining unconstrained features. Table 2 compares standard and anchored training across ReLU, JumpReLU, TopK, BatchTopK, and Matryoshka SAEs, as well as transcoders and crosscoders.

Feature anchoring improves feature recovery across the evaluated SDL families. For TopK-family methods, GT Recovery increases from 84.90% to 87.63% for TopK SAE, from 84.80% to 89.38% for BatchTopK SAE, and from 83.70% to 87.32% for Matryoshka SAE. ReLU and JumpReLU SAEs remain below the strict $\tau = 0.95$ recovery threshold, but their Maximum Inner Product increases substantially, indicating that anchoring improves alignment even when the recovery threshold is not reached.

Figure 9 reports GT Recovery over a range of matching thresholds. The improvement therefore does not arise solely from the particular $\tau = 0.95$ cutoff used in Table 2.

H.4 Feature Anchoring on Real CLIP Representations

We next test whether the same principle extends beyond synthetic representations where exact ground-truth feature directions are available. We extract CLIP ViT-B/32 (Radford et al., 2021) image embeddings for the ImageNet-1K training set (Russakovsky et al., 2015). Because the true semantic directions are unknown, we use normalized ImageNet class means as proxies:

$$\tilde{\mathbf{w}}_p^{(c)} = \frac{\bar{\mathbf{x}}_p^{(c)}}{\|\bar{\mathbf{x}}_p^{(c)}\|_2}, \tag{74}$$

where $\bar{\mathbf{x}}_p^{(c)}$ denotes the mean CLIP representation of images belonging to class c. We train TopK, BatchTopK, and Matryoshka SAEs with $n_p = n_r = 768$, $n_q = 16{,}384$, $k = 30$ anchors, and $\lambda_{\text{anchor}} = 1.0$.

Table 2: **Feature recovery on the Linear Representation Bench.** We use $n = 1000$, $n_p = n_r = 768$, $n_q = 16000$, and $\mathcal{S} = 0.99$. Evaluation excludes features used as anchors. Feature anchoring improves GT Recovery ($\mathbf{M}_{\mathrm{GT}}$) and/or Maximum Inner Product ($\mathbf{M}_{\mathrm{IP}}$) across the evaluated SDL methods. *For some methods, $\mathbf{M}_{\mathrm{GT}} = 0$ because all best-match similarities fall below the strict evaluation threshold $\tau = 0.95$; $\mathbf{M}_{\mathrm{IP}}$ remains informative in this regime.

Method	$\mathbf{M}_{\mathrm{GT}} \uparrow$	$\mathbf{M}_{\mathrm{IP}} \uparrow$
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**

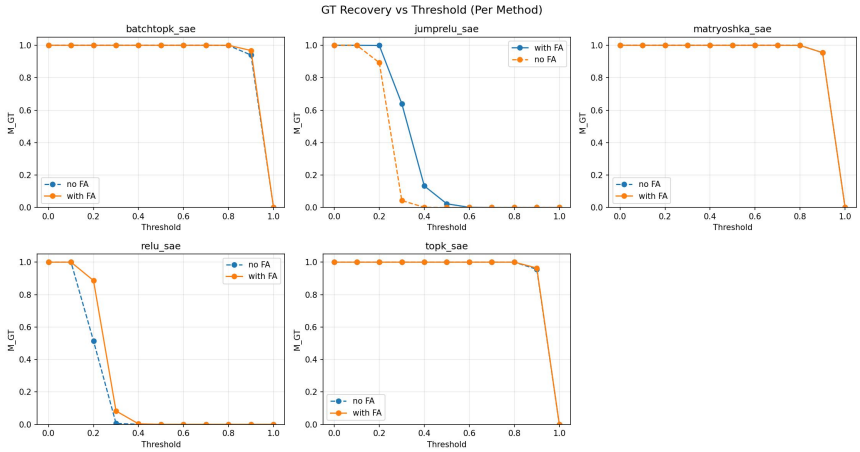

Figure 9: **GT Recovery across similarity thresholds.** Feature anchoring generally improves $\mathbf{M}_{\mathrm{GT}}$ across recovery thresholds, showing that the observed improvement is not specific to a single evaluation cutoff.

As shown in Table 3, standard SAEs do not recover the proxy semantic directions above the strict $\tau = 0.95$ threshold, whereas feature anchoring raises GT Recovery to approximately 24% for all three architectures. Maximum Inner Product also increases substantially, from 0.517 to 0.851 for TopK SAE, from 0.508 to 0.847 for BatchTopK SAE, and from 0.683 to 0.858 for Matryoshka SAE. Since evaluation excludes the explicitly anchored features, these results indicate that partial semantic constraints can influence the broader learned dictionary.

Table 3: **Feature recovery on CLIP representations.** Since ground-truth semantic directions are unavailable, normalized ImageNet class means are used as proxies. Evaluation excludes the explicitly anchored proxy directions. Feature anchoring substantially improves alignment with these semantic directions across TopK-family SAEs. *Zero GT Recovery indicates that no best-match similarities exceed $\tau = 0.95$.

Method	$\mathbf{M}_{\mathrm{GT}} \uparrow$	$\mathbf{M}_{\mathrm{IP}} \uparrow$
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**

H.5 Qualitative Examples on CLIP

The quantitative improvements are also reflected in the semantic coherence of learned features. Figure 10 shows representative features from a Matryoshka SAE trained with feature anchoring. Individual features correspond to coherent concepts such as “African Grey,” “American Black Bear,” and “Digital Clock,” with their top-activating images consistently depicting the corresponding semantic category.

In contrast, Figure 11 shows representative features learned by the same architecture without feature anchoring. These features frequently combine heterogeneous examples: one responds to several box-like objects, including dishwashers, file cabinets, and chests, while another responds to different types of screens, including slot machines, scoreboards, and televisions. These examples illustrate the ambiguity motivating our theoretical analysis: reconstruction quality alone does not guarantee a semantically aligned dictionary.

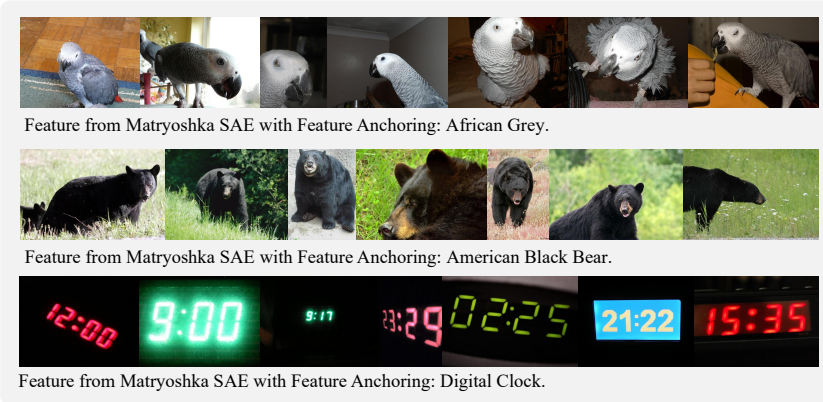

Figure 10: **Qualitative examples with feature anchoring.** Each row shows the top-activating images for a feature learned by a Matryoshka SAE on CLIP representations. The examples exhibit coherent semantic selectivity.

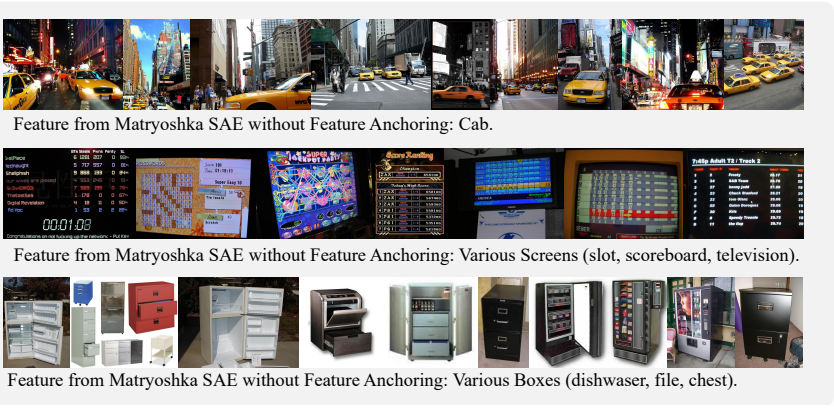

Figure 11: **Qualitative examples without feature anchoring.** Each row shows the top-activating images for a feature learned by the same Matryoshka SAE architecture without anchoring. Representative features respond to multiple heterogeneous concepts.

H.6 Ablation Studies

We further evaluate feature anchoring across different properties of the synthetic representation and SDL training procedure. We vary the degree of superposition, maximum feature interference, representation sparsity, and the TopK activation parameter. These experiments test whether the effect of anchoring is specific to the default benchmark configuration or persists across qualitatively different optimization regimes.

Effect of Superposition Ratio. We first vary the number of ground-truth features while keeping the representation dimension fixed. We train ReLU SAEs with $n_q = 16{,}000$ latent dimensions and $n \in \{800, 900, 1000, 1100, 1200, 1300, 1400\}$ ground-truth features while fixing $n_p = n_r = 768$. This produces superposition ratios n/n_p ranging from $1.04\times$ to $1.82\times$. Feature anchoring uses 100 randomly selected ground-truth anchors with $\lambda_{\text{anchor}} = 0.1$.

Table 4 shows that the benefit of feature anchoring is strongest when the number of superposed features is relatively close to the representation dimension. GT Recovery reaches 100% at $n = 800$ and 98.89% at $n = 900$, despite anchoring only a minority of ground-truth directions. Recovery decreases progressively as the degree of superposition increases, reaching 8.57% at $n = 1400$. Maximum Inner Product nevertheless improves under anchoring at every evaluated feature count. This trend is consistent with increased ambiguity as more ground-truth features are represented in a fixed-dimensional space, although our theoretical results do not by themselves imply this monotonic dependence on the superposition ratio.

Effect of Maximum Interference. We next vary the geometric overlap between ground-truth feature directions. We fix $n = 1000$ and $n_p = n_r = 768$ and control

$$M := \max_{i \neq j} \langle \mathbf{w}_p^i, \mathbf{w}_p^j \rangle, \tag{75}$$

which measures the maximum pairwise positive interference between feature directions.

Table 5 shows that TopK, BatchTopK, and Matryoshka SAEs consistently achieve higher GT Recovery with feature anchoring across all tested interference levels. For example, at $M = 0.5$, BatchTopK SAE improves from 83.7% to 89.6%. ReLU and JumpReLU SAEs remain below the strict recovery threshold but consistently exhibit higher Maximum Inner Product after anchoring. Thus, the effect of feature anchoring is not restricted to a particular degree of separation between the ground-truth directions.

Effect of Feature Sparsity. We vary the feature sparsity

$$\mathcal{S} = \Pr_{s \sim \mathcal{D}}(x_i(s) = 0), \tag{76}$$

Table 4: **Feature recovery under varying superposition ratios.** We fix $n_p = n_r = 768$ and vary the number of ground-truth features from 800 to 1400. Feature anchoring uses 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**

so that larger $\mathcal{S}$ corresponds to a lower feature activation rate and therefore a sparser representation.

At the default setting $\mathcal{S} = 0.99$, feature anchoring improves all evaluated methods in Maximum Inner Product and improves GT Recovery for the TopK-family methods. Under even more extreme sparsity, $\mathcal{S} = 0.995$, anchoring produces a particularly large improvement for JumpReLU SAE, increasing GT Recovery from 11.8% to 74.8%. At the denser setting $\mathcal{S} = 0.9$, TopK-family methods already recover nearly all ground-truth directions, and anchoring provides little additional benefit. These results indicate that the value of anchoring depends on the interaction between the activation mechanism and feature distribution rather than on sparsity alone.

Effect of the TopK Sparsity Parameter. Finally, we vary the TopK activation parameter $k_{\text{TopK}} \in \{32, 64, 128\}$, corresponding to activation rates of approximately 0.2%, 0.4%, and 0.8% of the $n_q = 16{,}000$ latent dimensions. We denote the activation parameter by k_{TopK} to distinguish it from the number of anchor features.

As shown in Table 7, both standard and anchored TopK SAEs achieve perfect GT Recovery at $k_{\text{TopK}} = 32$. At $k_{\text{TopK}} = 64$, both methods remain above 93% recovery, with no improvement from anchoring. At $k_{\text{TopK}} = 128$, anchoring again becomes beneficial, raising GT Recovery from 86.7% to 88.0%. These results suggest that sufficiently restrictive TopK activation can itself strongly constrain the SDL solution space, reducing the marginal benefit of additional semantic anchors.

Summary. Across these experiments, feature anchoring provides a concrete demonstration of how additional structural information can reduce ambiguity in SDL optimization. Its benefit is not uniform: when the activation mechanism already imposes strong constraints on the learned representation, as in highly effective TopK settings, anchoring may provide only marginal gains. In more underconstrained regimes, however, encouraging even a small subset of learned directions to align with semantic references can substantially alter the learned dictionary and improve recovery of features that were never directly anchored. We therefore view feature anchoring not as a separate contribution required by the theoretical framework, but as a theory-guided extension illustrating a practical way to mitigate the underdetermination characterized in the main paper.

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

Method	M	**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**

Table 6: **Feature recovery under varying feature sparsity.** Sparsity is defined as $\mathcal{S} = \Pr_{s\sim\mathcal{D}}(x_i(s) = 0)$, so larger $\mathcal{S}$ corresponds to sparser activations. All experiments use $n = 1000$, $n_p = n_r = 768$, and 100 anchors.

Method	Activation Rate $1-\mathcal{S}$	$\mathcal{S}$	GT Recovery ↑	Max Inner Product ↑
Extreme Sparsity ($\mathcal{S} = 0.995$)				
ReLU SAE	0.005	0.995	0.0%	0.231
+ Feature Anchoring	0.005	0.995	0.0%	**0.326**
JumpReLU SAE	0.005	0.995	11.8%	0.890
+ Feature Anchoring	0.005	0.995	**74.8%**	**0.970**
TopK SAE	0.005	0.995	82.2%	0.978
+ Feature Anchoring	0.005	0.995	**84.3%**	**0.981**
BatchTopK SAE	0.005	0.995	80.5%	0.976
+ Feature Anchoring	0.005	0.995	**83.2%**	**0.980**
Matryoshka SAE	0.005	0.995	79.3%	0.974
+ Feature Anchoring	0.005	0.995	**86.1%**	**0.983**
Default Sparsity ($\mathcal{S} = 0.99$)				
ReLU SAE	0.01	0.99	0.00%	0.205
+ Feature Anchoring	0.01	0.99	0.00%	**0.246**
JumpReLU SAE	0.01	0.99	0.00%	0.237
+ Feature Anchoring	0.01	0.99	0.00%	**0.333**
TopK SAE	0.01	0.99	84.90%	0.983
+ Feature Anchoring	0.01	0.99	**87.63%**	**0.986**
BatchTopK SAE	0.01	0.99	84.80%	0.981
+ Feature Anchoring	0.01	0.99	**89.38%**	**0.988**
Matryoshka SAE	0.01	0.99	83.70%	0.982
+ Feature Anchoring	0.01	0.99	**87.32%**	**0.985**
Moderate Sparsity ($\mathcal{S} = 0.9$)				
ReLU SAE	0.1	0.9	0.0%	0.186
+ Feature Anchoring	0.1	0.9	0.0%	**0.191**
JumpReLU SAE	0.1	0.9	0.0%	0.191
+ Feature Anchoring	0.1	0.9	0.0%	**0.193**
TopK SAE	0.1	0.9	**98.8%**	**0.981**
+ Feature Anchoring	0.1	0.9	98.4%	0.979
BatchTopK SAE	0.1	0.9	95.6%	0.975
+ Feature Anchoring	0.1	0.9	**96.2%**	**0.976**
Matryoshka SAE	0.1	0.9	99.7%	0.991
+ Feature Anchoring	0.1	0.9	99.7%	0.991

Table 7: **Feature recovery under varying TopK activation parameter.** All experiments use $n = 1000$, $n_p = n_r = 768$, $n_q = 16{,}000$, and 100 anchor features.

Method	k_{TopK}	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**