From Isolation to Entanglement: When Do Interpretability Methods Identify and Disentangle Known Concepts?

Aaron Mueller[1], **Andrew Lee**[2], **Shruti Joshi**[3], **Ekdeep Singh Lubana**[4], **Dhanya Sridhar**[3], **Patrik Reizinger**[5]
[1]Boston University [2]Harvard University [3]Mila – Quebec AI Institute
[4]Goodfire [5]University of Tübingen

Abstract

A goal of interpretability is to recover disentangled representations of latent concepts (features) from the activations of neural networks. The quality of features is typically evaluated in isolation, and under implicit independence assumptions that may not hold in practice. Thus, it is unclear to what extent common featurization methods such as sparse autoencoders (SAEs) and probes *disentangle* one concept from another. We propose a multi-concept evaluation setting using concepts such as sentiment, domain, voice, and tense. We evaluate how well featurizers produce disentangled representations of each concept, observing that features are typically sensitive to only one concept, but also that concepts are distributed across many features. Then, we steer these features, measuring whether each concept is independently manipulable, and whether features interact. Even in idealized settings, steering a feature often affects *many* concepts, despite a near absence of interaction effects. These results suggest that correlational metrics are insufficient to establish steering selectivity, and that demonstrating that two features operate in separate spaces is insufficient to claim that they will be selective for one concept. These results underscore the importance of multi-concept evaluations in interpretability research.[1]

1 Introduction

Interpretability centers on understanding and controlling neural network behaviors (Geiger et al., 2025; Mueller et al., 2025a). This requires understanding the underlying causal variables and mechanisms that produce observed input–output behaviors. To precisely localize these causal variables, *featurization methods*, such as sparse autoencoders (SAEs; Olshausen and Field, 1997; Bricken et al., 2023; Huben et al., 2024), have become common. These methods map from activation vectors (wherein a dimension can have many meanings) to sparser spaces where there is a more one-to-one relationship between dimensions and concepts.

The implicit assumption underlying these applications is that if we can identify features that represent distinct concepts, then we should be able to steer those concepts by by manipulating their corresponding features. But does representational disentanglement guarantee independent manipulability? Current concept identification and steering studies focus on detecting and/or steering single concepts or behaviors (e.g., Wu et al., 2025; Arditi et al., 2024; Marks and Tegmark, 2024). This tells us whether the concept is represented and can be manipulated, but leaves open the question of whether the concept representation is **independent** and **disentangled** from other concepts. How often does steering one concept affect others? Independence and disentanglement act as a ceiling for our trust in steering methods to induce similar behaviors in novel contexts—i.e., to what degree we have predictive power and selective control over the model's future behaviors.

This is not a new idea: the fields of causal representation learning (CRL; Schölkopf et al., 2021) and disentangled representation learning (Higgins et al., 2018; Locatello et al., 2019, 2020b) have rich literatures characterizing the assumptions under which it is possible to identify the true latent causal variables for a task. However, these fields focus on learning a representation from scratch, whereas the goal of interpretability is to derive a simplified causal model of a large and complex neural network that has already been trained. Both lines of work are unified in asking: *what methods and assumptions will yield causally efficacious representations?*

Our work builds upon and extends the metrics and evaluation paradigms of CRL to measure mechanistic disentanglement in a multi-concept evaluation setting. We design a dataset where each example has multiple ground-truth concept labels. We generate a natural language dataset using probabilistic context-free grammars (PCFG; Booth and Thompson, 1973), where each sentence is labeled with four concepts (voice, tense, sentiment, domain), and where we can control the degree of correlation between concepts in the dataset. We use this data to evaluate common interpretability methods, including probes (Gurnee et al., 2023) and sparse autoencoders (SAEs; Olshausen and Field, 1997; Huben et al., 2024), through two lenses: (1) Do sparse features and probes achieve high scores on standard correlational disentanglement metrics from CRL (MCC, DCI-ES)? And (2) when we steer features, do they selectively manipulate their target concepts without affecting others

[1]Data and code are available at https://github.com/aaronmueller/IdentifiableLanguage.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 17188–17210

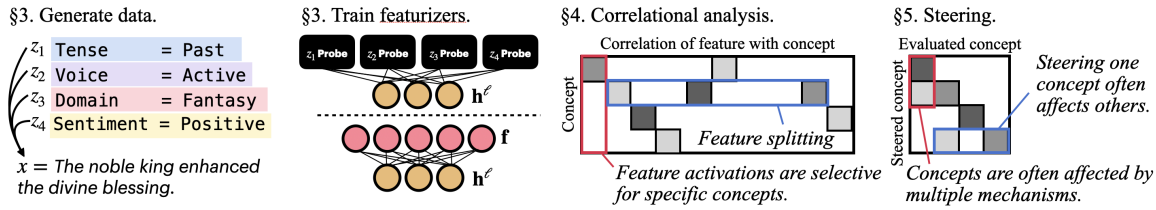

Figure 1: Overview. We first generate data where each example is labeled for multiple concepts (§3, App. A). Using this data, we train SAEs and probes (§3, App. B). We use correlational metrics to measure how well these representations disentangle each concept, finding that they are largely disentangled (§4), but also that SAEs exhibit feature splitting (§4.2). However, steering the top feature for a given concept often has significant effects on *many* concepts, suggesting entangled mechanisms (§5).

(independence), nor each other (disjointness)?

Our contributions and findings are:

- We introduce an evaluation framework and dataset to measure concept disentanglement, with adjustable correlations between ground-truth concepts (§3).
- We show that common SAE architectures and sparse probes achieve high disentanglement according to correlational metrics (§4.1), but that this does not predict the selectivity of steering (§5). We propose metrics to quantify concept entanglement in steering (§5.1).
- We distinguish between feature disjointness (having non-interacting effects) and selective steerability. Current methods produce features with non-interacting effects, but steering one feature nonetheless affects multiple unrelated concepts downstream (§5.2).

Our results suggest that even principled correlational metrics do not predict steering performance. Furthermore, disjointness and independence are not mutually predictive, suggesting that current featurization objectives may be optimizing for the wrong notions of concept separation. Overall, these findings underscore the importance of multi-concept evaluations, and separating correlational from interventional evidence.

2 Preliminaries

Methods. *Featurization* refers to a mapping from model representations $\mathbf{h}^\ell$ to some feature space $\mathbf{f}$ that is more interpretable. We focus on two classes of featurization methods: sparse autoencoders (SAEs) and linear probes. SAEs learn an encoder $\mathcal{F}$ that maps from a representation vector $\mathbf{h}^\ell$ to a higher-dimensional feature vector $\mathbf{f}$. It also learns a decoder $\mathcal{F}^{-1}$ that reconstructs $\mathbf{h}^\ell$ from $\mathbf{f}$.[2] The SAE is trained to reconstruct $\mathbf{h}^\ell$ accurately while also ensuring that $\mathbf{f}$ is sparse. We compare various SAE architectures; see Appendix B for details. Binary linear probes learn a linear transformation from $\mathbf{h}^\ell$ to a logit; if well-trained, this logit should correlate with the probability of the concept that the probe is trained to classify.

Metrics. The Mean Correlation Coefficient (**MCC**; Hyvarinen and Morioka, 2016) is a commonly used metric (Hyvarinen et al., 2019; Khemakhem et al., 2020b,a; Wendong et al., 2023; von Kügelgen et al., 2021, 2023; von Kügelgen, 2024; Reizinger et al., 2024, 2023b,a; Gresele et al., 2021, *i.a.*) that measures how well a representation recovers latent ground-truth factors. Given a feature vector $\mathbf{f}$ and a set of concepts Z, one locates the dimension $\hat{\mathbf{f}}_i$ that has the highest correlation with concept $z_i \in Z$. The MCC gives the mean of these maximal correlations across concepts, and equals 1.0 when there exist features that identify each concept up to permutation and scaling.

DCI-ES (Eastwood et al., 2022, 2023) provides correlational metrics for more precisely characterizing how well a given representation disentangles concepts; we use the **D**isentanglement, **C**ompleteness, **I**nformativeness, and **E**xplicitness metrics. We first construct $R \in \mathbb{R}^{|\mathbf{f}| \times |Z|}$, where $|\mathbf{f}|$ is the dimensionality of the feature vector and $|Z|$ is the number of ground-truth concepts. Each entry $R_{i,j}$ encodes the importance of $\mathbf{f}_i$ for predicting z_j.[3] D and C are then defined as:

$$D_i = 1 + \sum_{j \in Z} R_{i,j} \log R_{i,j} \quad (1)$$

$$C_j = 1 + \sum_{i \in \mathbf{f}} R_{i,j} \log R_{i,j} \quad (2)$$

D_i is proportional to the entropy of the importance of feature $\mathbf{f}_i$ across concepts. D_i is high (near 1) if $\mathbf{f}_i$ is only predictive of one concept z_j. D_i is low (near 0.0) if $\mathbf{f}_i$ is equally predictive of all concepts. C_j is proportional to the entropy of the predictability of concept z_j per feature. C_j is high (near 1.0) when z_j correlates strongly with only one feature's activations, and low (near 0.0) when correlated with many features.

I is the negative prediction error of a classifier trained on the feature vector $\mathbf{f}$: $I_j = 1 - \mathbb{E}_{x \in \mathcal{T}}[\mathcal{L}(\mathbf{f}, z_j)]$. This measures whether concept z_j can be recovered from the feature vector at all. We measure this by training linear probes on feature vectors. E measures how easily

[2] This is typically not a literal inversion. In SAEs, the decoder is learned such that the reconstruction error is minimized, but information is nonetheless lost when reconstructing $\mathbf{h}^\ell$.

[3] Following Eastwood and Williams (2018), these are derived by training classifiers on $\mathbf{f}$ to predict z_j. We use impurity-based feature importances. Importances across features sum to 1 for a given z_j.

concepts are recovered from the feature vector $\mathbf{f}$, proportional to the area under the loss-capacity curve. If all concepts are predictable with high accuracy using low-capacity probes, E is maximized; if high-capacity probes are needed, E is low.

Together, these metrics capture how easily each concept can be recovered from a feature space, whether a concept is split across features, and whether a feature is selective for only one concept. One might hypothesize that perfect scores on each would imply the ability to steer concepts in a modular way. As we show, this is not necessarily true.

3 Experimental Setup

Data. Our goal is to stress-test featurization methods by creating a dataset labeled with known concepts, but where concepts can be correlated to varying degrees. Using a probabilistic context-free grammar (PCFG), we generate a training dataset $\mathcal{D}$ containing 382,459 sentences and test dataset $\mathcal{T}$ consisting of 1,007 sentences, where each sentence is labeled for 4 concepts $z_i \in Z$: voice, tense, sentiment, and domain. In our datasets, voice (active, passive) and tense (present, past) are binary. Sentiment (positive, neutral, negative) is multinomial and ordinal, while domain (news, science, fantasy, other) is multinomial with no inherent ordering.

To create a less idealized setting, we fix correlations between two concept values—for example, positive sentiment and the science domain. We control correlations by upsampling examples where the concept values co-occur in $\mathcal{D}$ while training featurizer $\mathcal{F}$. Under varying correlational conditions, we observe to what extent the featurizer identifies the latent ground-truth concepts. See App. A for further details on data generation and example sentences.

Models and featurizers. A featurizer consists of an encoder $\mathcal{F} : \mathbb{R}^{|\mathbf{h}|} \to \mathbb{R}^{|\mathbf{f}|}$ and optionally a decoder $\mathcal{F}^{-1} : \mathbb{R}^{|\mathbf{f}|} \to \mathbb{R}^{|\mathbf{h}|}$. The encoder $\mathcal{F}$ maps hidden representation vector $\mathbf{h}^\ell$ at layer ℓ to features $\mathbf{f}$. We focus primarily on unsupervised methods such as sparse autoencoders (SAEs), due to their popularity in recent unsupervised interpretability research (Costa et al., 2025; Huben et al., 2024; Mueller et al., 2025a; Marks et al., 2025). We formally define each SAE architecture we test in App. B. We compare these methods to a supervised method, linear probing.

We use two models: Pythia-70M (Biderman et al., 2023) and Gemma-2-2B (Team et al., 2024). Their parameters are frozen in all experiments. We choose these because there exist publicly available SAEs trained on large natural language corpora, including the ReLU SAEs of Marks et al. (2025) for Pythia and JumpReLU SAEs of Lieberum et al. (2024) for Gemma.

The feature vector $\mathbf{f}$ should ideally encode one concept per dimension, regardless of the correlations between concepts in the training data.[4] Recent work has demonstrated the importance of the featurizer's inductive bias in ensuring this property, especially when deploying unsupervised featurizers (Hindupur et al., 2025; Costa et al., 2025). We therefore compare SAEs that make varying assumptions: ReLU SAEs (Bricken et al., 2023) assume linear separability, Top-K SAEs (Gao et al., 2025) assume angular separability, and SpADE SAEs (Costa et al., 2025) make weaker assumptions that allow for more heterogeneous concept geometries. SSAEs (Joshi et al., 2025) are trained on activation *differences* between pairs of inputs where concepts shift. We refer readers to App. B for details on each SAE architecture.

4 Features Represent Disentangled Concepts

4.1 Concept identification

A key desideratum of featurizers is the ability to identify the ground-truth concepts despite potential spurious correlations between them.[5] To assess to what degree this property holds for popular featurizers, we design an identifiability evaluation. Intuitively, identifiability measures whether and to what extent the learned model can recover the latent factors that generated the data (e.g., z_i in Figure 1). For formal definitions, see App. C.

Metrics. To evaluate whether a featurizer recovers ground-truth concepts, we first employ the **mean correlation coefficient** (MCC; Hyvarinen and Morioka, 2016) common in the causal representation learning literature. The MCC measures identifiability up to scaling and permutation (see §2 and App. D).

We correlate one feature in $\mathbf{f}$ with each concept. Howevre, multinomial concepts may not be one-dimensional in $\mathbf{f}$ nor $\mathbf{h}^\ell$ (Engels et al., 2025). Thus, we *binarize* concepts before computing MCC: given a concept z_i with $|z_i|$ possible values, we create a new binary variable for each value. For example, sentiment can take a value in {positive, neutral, negative}. We create one binary variable per value. To compute the MCC, we first average the correlation coefficients for all values of z_i before taking the macroaverage across concepts in Z. A high MCC is achievable in theory only if we make the following assumption:

Assumption: *For each value of each concept z_i, there exist linear transformations T_i such that $z_i = T_i\mathbf{h}^\ell$, where $\mathbf{h}^\ell$ are the representations of a language model at layer ℓ.*

To validate this assumption, we train linear probes for each binary concept value and observe whether each probe obtains high accuracy on the concept value it was trained to detect, *but also* obtains random-chance

[4] This is theroetically possible to learn as long as there are at least two examples where the concepts do not covary.

[5] We cannot expect a model, supervised or unsupervised, to be able to disentangle two concepts if they are *completely* correlated in the data (Wiedemer et al., 2023) without making any assumptions. However, given at least a couple examples where two concepts do not covary, it is possible in theory to recover independent representations of these concepts.

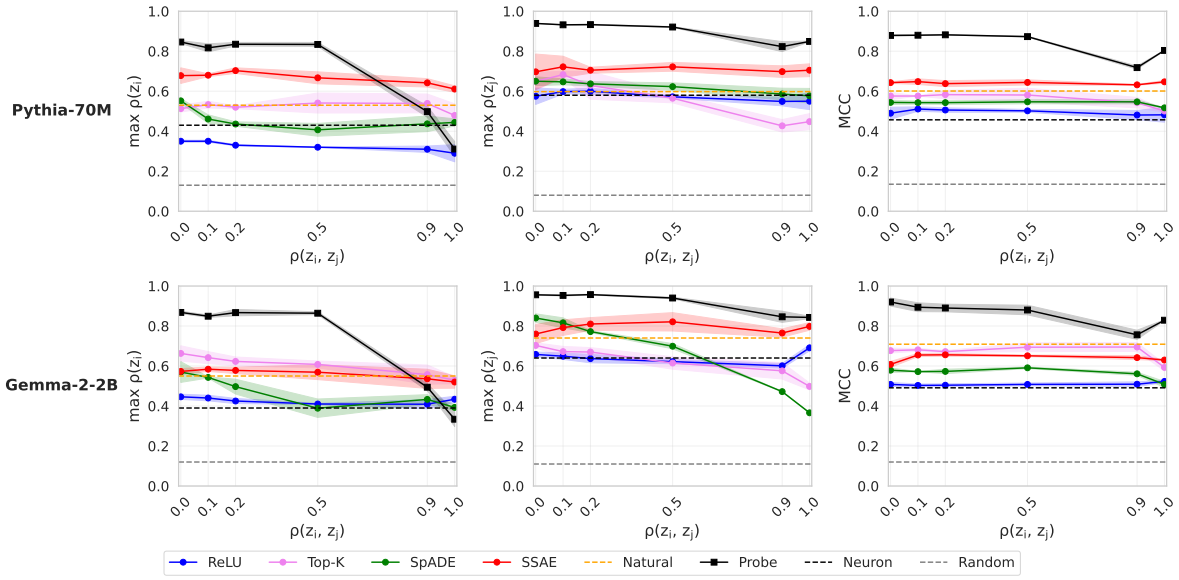

Figure 2: **Maximum correlation coefficient for domain=science (left), sentiment=positive (middle), and MCC (right) under varying correlational conditions**. Shaded regions represent 1 std. dev. across 3 training seeds. Ideal performance looks like a flat line at a high MCC. Probes (black) perform best up to cross-concept correlations of 0.9. SSAEs (red) and Top-k SAEs (pink) perform best among unsupervised featurizers. SAEs trained on large-scale natural data (Natural) perform similarly to our best SAEs, but SSAEs sometimes outperform both.

accuracy on all other concepts. Our probes satisfy these criteria and thus empirically support this Assumption; see Figure 9 (App. F).

Baselines and skylines. We compare against a randomly initialized SAE (*Random*); the original activation vector $\mathbf{h}^\ell$ (*Neuron*, equivalent to an identity featurizer $\mathbf{f} = \mathbf{h}^\ell$); and publicly available SAEs trained on natural language data (*Natural*; we use the SAEs of Marks et al. (2025) for Pythia-70M and GemmaScope (Lieberum et al., 2024) for Gemma-2-2B, respectively).

To establish a supervised skyline (*Probe*), we train binary logistic regression probes for each concept value (i.e., we have separate binary probes for negative sentiment, positive sentiment, past tense, etc.). We correlate each probe's logit with ground-truth concept labels, and take the average correlation across concepts to compute the MCC.

Hypothesis. The ideal result is a high MCC that remains constant as the correlation between ground-truth concepts increases in the training data. Among (S)SAEs, we hypothesize that SSAEs will perform best, as sparse representations of *shifts* between concepts are provably identifiable, whereas typical SAE architectures do not have this property (Joshi et al., 2025). We expect unsupervised featurizers, such as SAEs, to perform worse than supervised featurizers, such as probes. We also expect SAEs trained on our dataset to be better able to isolate the ground-truth concepts compared to the *Natural* baselines; this is because the number of varying concepts is lower, which should make these concepts easier to isolate.

Results. Figure 2 shows MCCs, as well as the maximum correlation coefficients for Pythia-70M and Gemma-2-2B for the domain and sentiment concepts as they become more correlated in the training dataset. Probes significantly outperform SAEs, as expected (up to correlations of 0.9). The margin between probes and SAEs is generally substantial; thus, if one knows *a priori* what concepts one wishes to find, one should use supervised methods. This agrees with recommendations from Wu et al. (2025) and Mueller et al. (2025b).

SSAEs perform best or close to best among (S)SAEs, as hypothesized. Top-K SAEs also perform well, although they underperform for sentiment=positive. Our SAEs trained on synthetically generated data achieve comparable performance to SAEs trained on a much larger natural language corpus (the *Natural* SAEs in Figure 2); SSAEs outperform both for Pythia-70M, but not for Gemma-2-2B. Other methods achieve comparable or lower performance. Thus, to locate concepts in language models, *one may not need to worry about curating concept-specific data if one's dataset is sufficiently large.* But also, the SAE architecture makes a significant difference.

When do correlations between concepts start to impede concept identification? The answer depends on the method: probes and SpADE (Costa et al., 2025) maintain consistent MCCs up to correlations of 0.5 between concept pairs in the training data. Beyond this, concept representations degrade. For SSAEs, MCC remains consistent up to complete correlations of 1.0, as its theory predicts (Joshi et al., 2025). Overall, these results suggest that if one uses an optimal architecture, one may not need to be concerned about spurious correlates of a

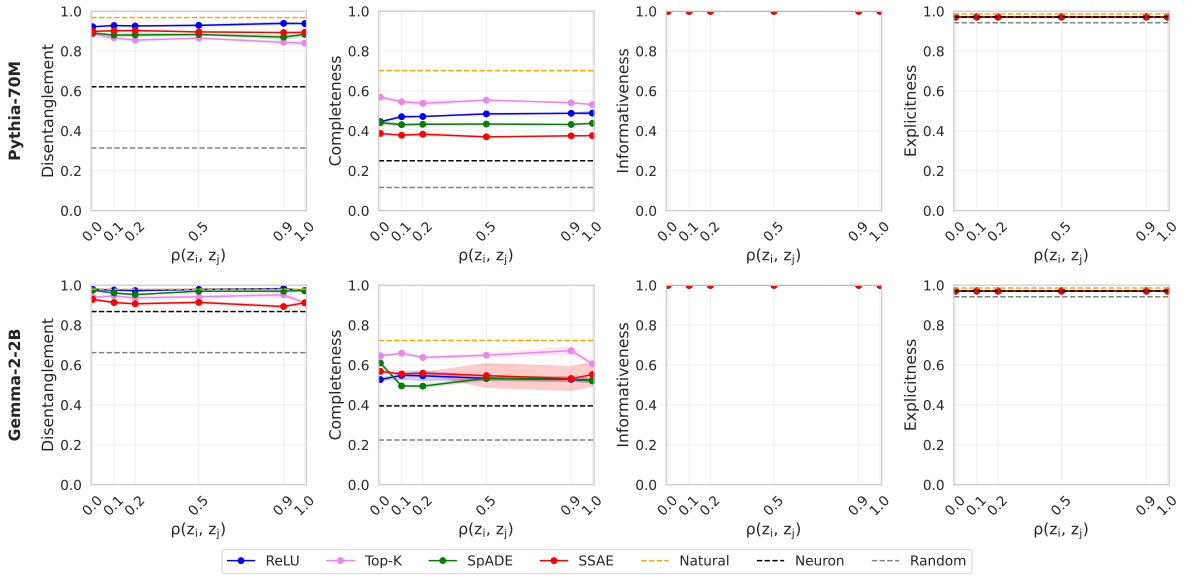

Figure 3: **DCI-ES scores under varying correlational conditions**. Shaded regions represent 1 std. dev. across 3 training seeds. Ideal performance looks like a flat line at 1 for all metrics. All methods achieve high disentanglement, informativeness, and explicitness, but relatively low completeness. This suggests that most features capture only one concept, but also that concepts are generally distributed across multiple features.

target concept, unless that correlation is near-complete.

4.2 Diagnosing Failures in Identifiability

As seen in §4.1, MCCs are far from ideal. To diagnose the cause, we conduct a more fine-grained evaluation.

Metrics. We use the DCI-ES framework of Eastwood et al. (2023). Specifically, we use the **D**isentanglement, **C**ompleteness, **I**nformativeness, and **E**xplicitness metrics. See §2 and App. D for intuitive and detailed definitions, respectively. In this setting, DCI-ES allows us to diagnose phenomena such as *feature splitting* (low completeness), *concept entanglement* (low disentanglement), the features not encoding the concept at all (low informativeness), or features encoding concepts in a difficult-to-recover way (low explicitness).

DCI-ES can indicate whether and to what extent (or equivalence class) identifiability is achieved. Identifiability up to *invertible linear transformations* is achieved if $I = E = 1$; up to permutation and element-wise reparametrization if $D = C = I = 1$; and *up to sign and permutation* if $D = C = I = E = 1$. Importantly, steering is not guaranteed to work when $I = E = 1$, as for steering, we select the single most correlated dimension, which can be a linear mixture of multiple concepts. $D = C = I = E = 1$ implies that all concepts are encoded in a single feature, which means we could predict the impact of steering on concept probabilities via linear extrapolation—even under multiple steering operations.

Hypothesis. We hypothesize that all concepts will be recoverable from SAE feature vectors—i.e., that I will be near 1. Because SAEs are trained to be sparse, we expect E to be close to 1. We also hypothesize that the main failure mode will be feature splitting (that is, one concept being split across many features). If so, C should be low.

Results. We observe (Figure 3) that D, I, and E are high for all SAE architectures, but not for the original representation space of the models. This suggests that all concepts are near-perfectly recoverable (high I) with limited-capacity classifiers (high E), and that each SAE identifies the ground-truth concepts up to linear transformation. However, C is low, which suggests widespread feature splitting, and that the SAEs do not identify concepts up to sign and permutation. Most features are sensitive to one concept (high D), but concepts are often distributed across many features (low C).

To what degree does feature splitting occur? To quantify, we use k-sparse probes (Gurnee et al., 2023) and analyze how many features are necessary before probe performance saturates. This generally requires 10 features; see App. E.

High D, I, and E suggest that steering should only affect the probability of the target concept being steered (i.e., that features will generally be selective for one concept). In the following section, we test these predictions by steering the top SAE feature for each concept.

5 Steering as a Causal Independence Test

MCC and DCI-ES only provide correlational evidence. However, none of these metrics provide *causal* evidence that we can independently manipulate concepts using the learned features. Thus, to measure causal efficacy, we employ steering as a test of independence of the mechanisms associated with each feature. This can be seen as testing the Independent Causal Mechanism principle (Pearl, 2009; Peters et al., 2018), which holds

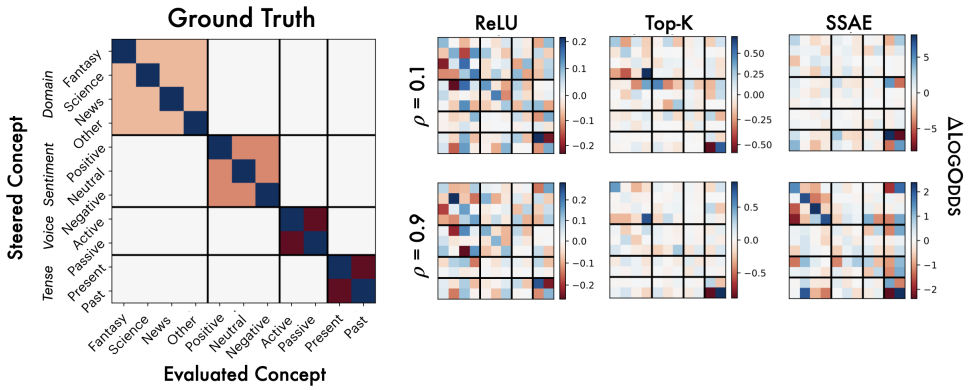

Figure 4: **The effect of steering a given concept (row) on the log-odds of another (column), as measured by a probe**. Results for Pythia-70M shown here; see App. I.1 for Gemma-2-2B. If concept representations are causally independent, we expect a heatmap that resembles the ground-truth: ΔLogOdds should be high on the diagonal, negative for within-concept pairs, and close to 0.0 for across-concept pairs. All SAEs demonstrate the expected diagonals, but also significant across-concept effects. Increasing correlations in the training data, even up to 0.9, yield qualitatively similar entanglement patterns.

that different causal mechanisms neither influence nor inform each other.

5.1 Independence and Selectivity

We steer the top feature $\hat{\mathbf{f}}_i$ for concept z_i. To locate $\hat{\mathbf{f}}_i$, we select the feature whose activation correlates most with z_i, as when computing the MCC in §4.1.[6]

Steering the activations $\mathbf{h}^\ell$ with feature $\mathbf{f}_i$ is performed using a steering function $\tilde{\mathbf{h}}^\ell(\mathbf{f}_i) \leftarrow \Phi(\mathbf{h}^\ell, \mathcal{F}, i, \alpha)$, where Φ is defined as follows:

$$\Phi(\mathbf{h}^\ell, \mathcal{F}, i, \alpha) = \mathcal{F}^{-1}\Big(\mathcal{F}(\mathbf{h}^\ell)|\text{do}(\mathbf{f}_i = \alpha \cdot \max(f_i))\Big) + \epsilon \tag{3}$$

α, the steering coefficient, controls the strength of the steering operation; $\mathcal{F}(\mathbf{h})$ corresponds to the featurized activations (equivalent to $\mathbf{f}$); and the do-operation denotes an intervention where feature $\mathbf{f}_i$ is set to α times its maximum activation $\max(f_i)$ on training dataset $\mathcal{D}$. $\epsilon = \mathbf{h} - \mathcal{F}^{-1}(\mathcal{F}(\mathbf{h}))$ is the reconstruction error before interventions; adding it to the steered output ensures that any changes in model behavior are due to the steering operation, and not due to reconstruction errors (Marks et al., 2025). We set α to 5, but try multiple coefficients in §5.2.

Metrics. For all concept pairs (z_i, z_j), we steer the feature most associated with z_i and measure ΔLogOdds of concept z_j. We quantify $\Delta\text{LogOdds}(z_j)$ as the change in the logit of z_j according to a multinomial concept probe trained on the *final* layer of the model.[7]

We introduce **concept independence** $\mathcal{I}_S$ to quantify to what degree a concept is influenced only by its top feature and no others, and **feature selectivity** $\mathcal{S}_S$ to quantify to what degree a feature only influences its respective concept. To measure both, we construct a matrix $S \in \mathbb{R}^{Z \times Z}$, where rows correspond to steering $\hat{\mathbf{f}}_i$, the top feature for concept i, and columns correspond to $\Delta\text{LogOdds}(z_j)$. Then, for concept z_i, $\mathcal{I}_S$ is $S_{i,i}$ divided by the sum over column i. $\mathcal{S}_S$ is defined as $S_{i,i}$ divided by the sum over row i. More formally:

$$\mathcal{I}_S = \frac{|\log p(z_i|\tilde{\mathbf{h}}^\ell(\hat{\mathbf{f}}_i)) - \log p(z_i)|}{\sum_{j \neq i}(|\log p(z_i|\tilde{\mathbf{h}}(\hat{\mathbf{f}}_j)) - \log p(z_i)|)}, \tag{4}$$

$$\mathcal{S}_S = \frac{|\log p(z_i|\tilde{\mathbf{h}}^\ell(\hat{\mathbf{f}}_i)) - \log p(z_i)|}{\sum_{j \neq i}(|\log p((z_j|\tilde{\mathbf{h}}(\hat{\mathbf{f}}_i)) - \log p(z_j)|)}, \tag{5}$$

In words, $\mathcal{I}_S$ is maximized when steering $\hat{\mathbf{f}}_i$ affects $\log p(z_i)$ far more than steering any $\hat{\mathbf{f}}_j$, the top features for other concepts z_j. $\mathcal{S}_S$ is maximized when steering $\hat{\mathbf{f}}_i$ affects $\log p(z_i)$ far more than it affects any $\log p(z_j)$. In both equations, j excludes all values of concept z_i. For example, if i is domain=news, j would skip all domains, including fantasy, news, etc.

Hypothesis. Because we observed high disentanglement in §4, we expect high feature selectivity and high concept independence. Because we observed low completeness, failures in steering should be because steering fails to affect its target concept, and not because a feature significantly affects unintended concepts.

Results. We observe (Figure 4) that for each SAE architecture, the expected diagonal trend is present, indicating that steering is increasing the log-odds of the

[6] However, Arad et al. (2025) has found that the features that *detect* the input concept (the top-correlated features in our case) and the features that *control* the concept in a model's outputs are nearly disjoint. Thus, we also test using gradient attributions (Simonyan et al., 2014) to locate the feature that should be steered. See App. I.2.

[7] We use the final layer because it acts as a better proxy for the model's likely output behavior, as opposed to the model's inner representation of the input concepts. We use multinomial probes because they make the change in probabilities for within-concept pairs sum to 1. To validate that multinomial probe logits are good proxies for concept presence, we show heatmaps of probe accuracies in Figure 10 (App. F).

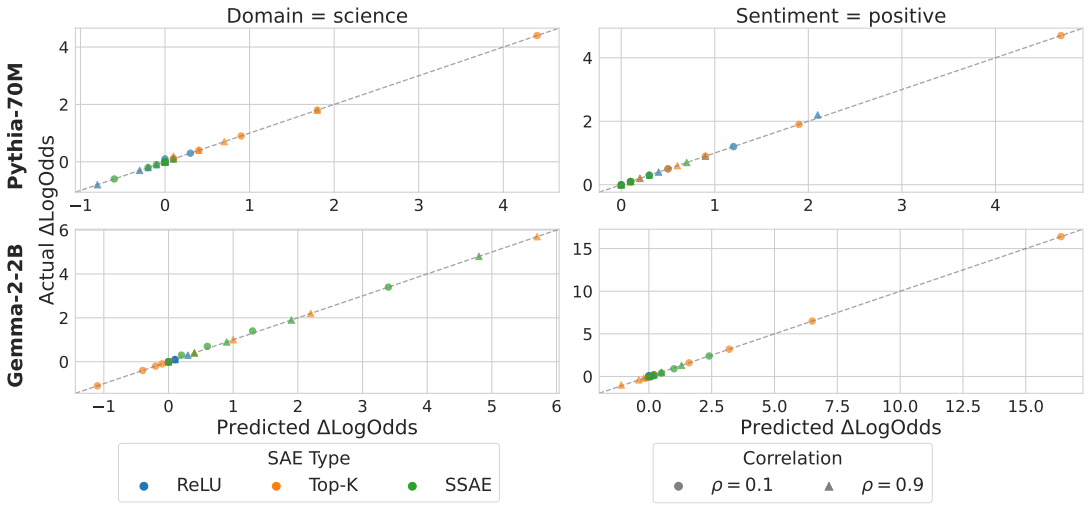

Figure 5: **Predicted ΔLogOdds(z_i) under disjointness assumptions vs. actual ΔLogOdds(z_i) when steering relevant feature $\hat{\mathbf{f}}_i$ and unrelated feature $\hat{\mathbf{f}}_j$.** Predicted ΔLogOdds are obtained by adding the ΔLogOdds(z_i) when steering with either $\hat{\mathbf{f}}_i$ or $\hat{\mathbf{f}}_j$ in separate forward passes. Actual ΔLogOdds values are obtained by steering both in the same forward pass. $\hat{\mathbf{f}}_i$ and $\hat{\mathbf{f}}_j$ are typically disjoint, as indicated by the predicted change almost perfectly matching the actual change.

		Pythia-70M		Gemma-2-2B	
SAE	ρ	Independence	Selectivity	Independence	Selectivity
ReLU	0.1	0.32 (**0.53**)	0.29 (**0.38**)	0.24 (**0.49**)	0.25 (**0.36**)
	0.9	0.30 (**0.40**)	0.31 (**0.41**)	0.23 (**0.62**)	0.29 (**0.52**)
Top-K	0.1	0.60 (**0.86**)	0.84 (**1.01**)	0.43 (**0.65**)	0.45 (**0.60**)
	0.9	0.49 (**0.55**)	0.53 (**0.76**)	0.76 (**1.25**)	0.67 (**0.85**)
SSAE	0.1	0.42 (**0.65**)	0.29 (**0.50**)	0.50 (**1.67**)	0.37 (**0.80**)
	0.9	0.37 (**0.71**)	0.50 (**1.29**)	0.72 (**1.21**)	0.36 (**0.83**)

Table 1: **Steering independence and steering selectivity scores.** We present mean scores per feature/concept, and maxima across features/concepts in parentheses and bold. High independence means that a concept is only influenced by one feature; high selectivity means that a feature only influences one concept. Mean independence and selectivity are generally low, indicating widespread entanglement; however, maximal scores are high, indicating that at least one concept is selectively recovered by these architectures.

target concept as expected. However, in even the best architectures, steering leads to measurable impacts on many unrelated concepts, indicating that selectivity is low. Moreover, concepts are often affected by many features, indicating widespread non-independence. Table 1 quantitatively summarizes these results. Best-case scores are high, but mean scores are lower, indicating that selective manipulability is achieved only for a subset of concepts.

This underscores the importance of both multi-concept evaluations *and* counterfactual interventions in evaluating concept representations: the correlational analyses did not suggest that interference would be likely in a steering setup, and yet we find evidence of widespread non-selectivity/non-independence. This may extend the findings of Arad et al. (2025), who claim that "input features" (those detecting a concept in the inputs) and "output features" (those controlling the model's use of the concept in its outputs) are disjoint. If this is true, then disentangled input features may by irrelevant to our ability to selectively steer.

5.2 Disjointness

Steering with one concept and evaluating across many others can provide causal evidence as to how disentangled two concept mechanisms are. Now, inspired by Zuheng et al. (2024), we ask whether these concept representations are **disjoint**—i.e., whether they affect non-overlapping subspaces, and have no non-linear interaction terms. This is not equivalent to independence nor selectivity:[8] even if two features have no interaction terms, each could still produce non-zero effects on multiple concepts.

Disjointness implies that we can predict the effect of pairs of steering operations on z_i from individual steering operations, even if individual steering operations affect multiple concepts. Studying disjointness is important because its presence gives us predictive power over model behavior, even on unseen or out-of-distribution data. See Figure 7 for illustrations and a direct contrast of independence and disjointness. Formally, disjointness is achieved when:

$$\log p(z_i|\tilde{\mathbf{h}}^\ell(\hat{\mathbf{f}}_i,\hat{\mathbf{f}}_j)) - \log p(z_i|\mathbf{h}^\ell) = \Big(\log p(z_i|\tilde{\mathbf{h}}^\ell(\hat{\mathbf{f}}_i)) - \log p(z_i|\mathbf{h}^\ell)\Big) + \Big(\log p(z_i|\tilde{\mathbf{h}}^\ell(\hat{\mathbf{f}}_j)) - \log p(z_i|\mathbf{h}^\ell)\Big). \quad (6)$$

[8] See Figure 7 (App. D) for an illustration and discussion of the difference between independence and disjointness.

SAE	ρ	Domain=sentiment		Sentiment=positive	
		Pythia-70M	Gemma-2-2B	Pythia-70M	Gemma-2-2B
ReLU	0.1	1.00	0.99	1.00	1.00
	0.9	1.00	1.00	1.00	1.00
Top-K	0.1	1.00	0.99	1.00	0.98
	0.9	1.00	1.00	1.00	0.99
SSAE	0.1	1.00	1.00	1.00	1.00
	0.9	1.00	1.00	1.00	1.00

Table 2: R^2 **between predicted and actual ΔLogOdds(z_i) for each SAE.** Values are all near 1.00, indicating near-perfect disjointness for each SAE, even under high correlations between concepts.

That is, the change in $\log p(z_i)$ when steering both $\hat{\mathbf{f}}_i$ and $\hat{\mathbf{f}}_j$ in one forward pass should be equivalent to the sum of the changes when steering $\hat{\mathbf{f}}_i$ and $\hat{\mathbf{f}}_j$ in separate forward passes. In practice, we again use ΔLogOdds rather than probabilities, as they are more likely to be additive at especially high and low probabilities due to greater numeric precision.

We try steering coefficients $\alpha \in \{0.1, 0.5, 1.0, 2.0, 5.0\}$. To compute predicted ΔLogOdds, we steer $\hat{\mathbf{f}}_i$ and $\hat{\mathbf{f}}_j$ in separate forward passes and sum their effects on the LogOdds of concept z_i. To compute actual ΔLogOdds, we steer both features in the same forward pass and measure the effect on z_i. If the prediction is equal to the actual change, we say that $\hat{\mathbf{f}}_i$ and $\hat{\mathbf{f}}_j$ are disjoint.

Hypothesis. Under low correlations, we expect that features will be disjoint, such that the effect of steering the top features for z_i and z_j on ΔLogOdds(z_i) will be additive, regardless of the concepts' (non-)independence. Under higher correlations between concepts in the data, we expect less disjoint concept representations.

Results. We observe (Figure 5) that the effect of steering with two concepts simultaneously is almost exactly equivalent to summing the impact of steering with both concepts separately. To quantitatively verify, we compute the R^2 between predicted and actual ΔLogOdds; these results (Table 2) suggest almost no interaction.

This in combination with the non-independence results of §5.1 suggests that feature interactions do not explain non-selective steering. This suggests that the independence of two concept mechanisms cannot be established by demonstrating that their component sets or subspaces do not overlap: one *must* observe changes to output behaviors before and after intervention.

6 Related Work

Featurization. In interpretability, *featurization* refers to techniques that allow one to map from less interpretable representations—typically activations—to more interpretable (and often sparser) representations (often called *features*). This has produced supervised techniques such as sparse probing (Gurnee et al., 2023), unsupervised techniques such as sparse autoencoders (SAEs; Olshausen and Field, 1997; Bricken et al., 2023; Huben et al., 2024), and non-parametric techniques such as steering vectors (Subramani et al., 2022) derived via difference-in-means (Marks and Tegmark, 2024).

How can one evaluate the quality of a feature? Recent work has proposed standardized evaluations based on known concepts (Mueller et al., 2025b; Huang et al., 2024; Wu et al., 2025). These allow one to assess whether a concept discovery method can represent or enable counterfactual manipulation of a concept with high recall. Such studies suggest that SAEs are generally worse than simple supervised approaches like probes and difference-in-means. While Wu et al. (2025) and Mueller et al. (2025b) consider the quality of steering methods for one concept at a time, our benchmark additionally considers the relationship between the steered concept and others. We propose that evaluating disentanglement in steering requires multi-concept evaluations.

Causal representation learning. Causal representation learning (CRL; Schölkopf et al., 2021) assumes that high-dimensional observations are generated from low-dimensional latent factors, whose relationships to other latent factors are encoded in a causal graph. Then, CRL proposes latent variable models of such observations that are **identifiable**, meaning that the recovered features are related to the true factors up to permutation and element-wise transformations. Because unsupervised learning is not identifiable without further assumptions (Hyvärinen and Pajunen, 1999; Darmois, 1951; Locatello et al., 2019), CRL methods rely on non-i.i.d. data or constraints on the decoding function (Moran et al., 2022; Gresele et al., 2021; Lachapelle et al., 2023b; Brady et al., 2025; Reizinger et al., 2023b). For example, CRL has developed identifiable models using data from sparse interventions (Ahuja et al., 2023a; Zhang et al., 2023; Buchholz et al., 2023; von Kügelgen et al., 2023), contrastive pairs of samples (Ahuja et al., 2022; Locatello et al., 2020a; Gresele et al., 2019; Brehmer et al., 2022), data from multiple environments (Ahuja et al., 2023b; Layne et al., 2025; Khemakhem et al., 2020a), and temporal data with sparse or intervened mechanisms (Lachapelle et al., 2021; Lippe et al., 2023, 2022). We go further, however, and test the causal implications of disentangled features on model outputs. Similar to what we propose, Joshi et al. (2025) propose a method that enables identifiable steering under multi-concept shifts; this method often performs well on disentanglement *and* steering-based metrics.

To corroborate the theoretical claims of identifiability, access to the ground-truth factors is required, which generally limits the tasks that can be considered. Among the evaluation metrics, the MCC (Hyvarinen and Morioka, 2016) has been used widely, despite known shortcomings (Hsu et al., 2023). Several other metrics have been proposed in both the disentanglement and the identifiable (causal) representation learning communities, such as the IRS score that measures interventional effects (Suter et al., 2019), or the DCI (Eastwood and Williams, 2018), DCI-ES (Eastwood et al., 2023), and

InfoMEC (Hsu et al., 2023) scores that directly aim to address shortcomings of the MCC. See the concurrent work of Joshi et al. (2026) for a detailed analysis.

7 Discussion and Conclusions

Our experiments reveal that current featurization methods, such as SAEs and sparse probes, show strong disentanglement when measured via correlational and representational evaluation metrics (§4.1). Further analyses reveal that any failures in identifiability are usually due to the existence of many features for one concept, and not due to entanglement of concepts in individual features (§4.2). We also observe improvements in sparsity and disentanglement over the native neuron-based representation space of a model.

Even so, steering experiments reveal that entanglement in the output space can be widespread (§5.1, 5.2), even when correlational measures of disentanglement suggest otherwise. This extends findings from Arad et al. (2025) to a multi-concept evaluation setting: disentangled input representations do not imply selective manipulability.

Despite the non-selectivity of features and non-independence of concepts during steering, most feature pairs demonstrate negligible interaction effects (§5.2) and operate over disjoint subspaces. This implies that when features achieve the *form* of separation—that is, that they have no non-linear interactions in the activation spaces they correspond to—this does not necessarily imply that their *functional roles* are non-interacting. This suggests that interpretability studies aiming to establish the independence of two mechanisms cannot settle for establishing that their subspaces or circuits do not overlap; one must establish independence by observing the output behaviors of a system before and after interventions to those mechanisms. This could imply that circuit overlap is a poor proxy for mechanistic independence, or that featurizers that optimize for feature orthogonality may not actually be recovering independent mechanisms.

Limitations

Our data is generated by a PCFG. While it is natural language, it is still a far narrower distribution of text compared to the distributions on which sparse autoencoders are normally trained or evaluated.

Relatedly, our dataset is relatively simple in that it considers four concepts. At a high level, we see two complementary research needs in this space: the first focuses on understanding why and under what conditions steering might fail, and the second measures how often features are likely to fail in practice. Our work is aligned with the first need, whereas larger-scale benchmarks like AxBench (Wu et al., 2025) and MIB (Mueller et al., 2025b) are aligned with the second. Our choice to focus on a smaller range of concepts allows us to precisely evaluate when and why concepts can be identified and steered by allowing us to create perfect ground-truth labels for many factors for each input. Nonetheless, we acknowledge that our conclusions could be strengthened by extending these experiments to a larger number of concepts. Relatedly, we also only study categorical concepts. Extending this framework to continuous concepts could reveal interesting new trends.

Acknowledgments

This work was initiated at the Fourth Bellairs Workshop on Causality, held at the McGill University Bellairs Research Institute (14–21 February, 2025). We are grateful to the organizers, Perouz Taslakian and Alexandre Drouin, for enabling this collaboration, and to the participants of the workshop for many thoughtful discussions during the ideation phase of this project. We thank Zhijing Jin, Vitória Barin Pacela, Victor Veitch, Atticus Geiger, Kartik Ahuja, Frederick Eberhardt, and Thomas Icard for extended discussions on earlier versions of these ideas. We also thank Sankaran Vaidyanathan for constructive feedback.

This work was supported by a grant from Coefficient Giving to Aaron Mueller. Patrik Reizinger acknowledges his membership in the European Laboratory for Learning and Intelligent Systems (ELLIS) PhD program and thanks the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for its support.

References

Kartik Ahuja, Jason S Hartford, and Yoshua Bengio. 2022. Weakly supervised representation learning with sparse perturbations. *Advances in Neural Information Processing Systems*, 35:15516–15528.

Kartik Ahuja, Divyat Mahajan, Yixin Wang, and Yoshua Bengio. 2023a. Interventional causal representation learning. In *International Conference on Machine Learning*, pages 372–407. PMLR.

Kartik Ahuja, Amin Mansouri, and Yixin Wang. 2023b. Multi-Domain Causal Representation Learning via Weak Distributional Invariances. *arXiv preprint*. ArXiv:2310.02854 [cs, stat].

Dana Arad, Aaron Mueller, and Yonatan Belinkov. 2025. SAEs are good for steering – if you select the right features. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 10241–10259, Suzhou, China. Association for Computational Linguistics.

Andy Arditi, Oscar Balcells Obeso, Aaquib Syed, Daniel Paleka, Nina Rimsky, Wes Gurnee, and Neel Nanda. 2024. Refusal in language models is mediated by a single direction. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai

Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. 2023. Pythia: a suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Taylor L. Booth and Richarda . Thompson. 1973. Applying probability measures to abstract languages. *IEEE Transactions on Computers*, C-22:442–450.

Jack Brady, Julius von Kügelgen, Sebastien Lachapelle, Simon Buchholz, Thomas Kipf, and Wieland Brendel. 2025. Interaction asymmetry: A general principle for learning composable abstractions. In *The Thirteenth International Conference on Learning Representations*.

Johann Brehmer, Pim de Haan, Phillip Lippe, and Taco Cohen. 2022. Weakly supervised causal representation learning. *arXiv preprint*. ArXiv:2203.16437 [cs, stat].

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, and 6 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Simon Buchholz, Goutham Rajendran, Elan Rosenfeld, Bryon Aragam, Bernhard Schölkopf, and Pradeep Ravikumar. 2023. Learning linear causal representations from interventions under general nonlinear mixing. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*.

Valérie Costa, Thomas Fel, Ekdeep Singh Lubana, Bahareh Tolooshams, and Demba E. Ba. 2025. From flat to hierarchical: Extracting sparse representations with matching pursuit. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

George Darmois. 1951. Analyse des liaisons de probabilité. In *Proc. Int. Stat. Conferences 1947*, page 231.

Cian Eastwood, Armin Kekic, and Andrei Liviu Nicolicioiu. 2022. On the DCI Framework for Evaluating Disentangled Representations: Extensions and Connections to Identifiability. page 8.

Cian Eastwood, Andrei Liviu Nicolicioiu, Julius Von Kügelgen, Armin Kekić, Frederik Träuble, Andrea Dittadi, and Bernhard Schölkopf. 2023. DCI-ES: An extended disentanglement framework with connections to identifiability. In *The Eleventh International Conference on Learning Representations*.

Cian Eastwood and Christopher K. I. Williams. 2018. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*.

Joshua Engels, Eric J Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. 2025. Not all language model features are one-dimensionally linear. In *The Thirteenth International Conference on Learning Representations*.

Jose Gallego-Posada and Juan Ramirez. 2022. Cooper: a toolkit for Lagrangian-based constrained optimization. `https://github.com/cooper-org/cooper`.

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

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, and Thomas Icard. 2025. Causal abstraction: A theoretical foundation for mechanistic interpretability. *Journal of Machine Learning Research*, 26(83):1–64.

Gauthier Gidel, Hugo Berard, Gaëtan Vignoud, Pascal Vincent, and Simon Lacoste-Julien. 2020. A variational inequality perspective on generative adversarial networks. *Preprint*, arXiv:1802.10551.

Luigi Gresele, Paul K. Rubenstein, Arash Mehrjou, Francesco Locatello, and Bernhard Schölkopf. 2019. The Incomplete Rosetta Stone Problem: Identifiability Results for Multi-View Nonlinear ICA. *arXiv:1905.06642 [cs, stat]*. ArXiv: 1905.06642.

Luigi Gresele, Julius von Kügelgen, Vincent Stimper, Bernhard Schölkopf, and Michel Besserve. 2021. Independent mechanism analysis, a new concept? *arXiv:2106.05200 [cs, stat]*. ArXiv: 2106.05200.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. 2023. Finding neurons in a haystack: Case studies with sparse probing. *Transactions on Machine Learning Research*.

Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. 2018. Towards a Definition of Disentangled Representations. *arXiv:1812.02230 [cs, stat]*. ArXiv: 1812.02230.

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

Kyle Hsu, Will Dorrell, James C. R. Whittington, Jiajun Wu, and Chelsea Finn. 2023. Disentanglement via Latent Quantization. *arXiv preprint*. ArXiv:2305.18378 [cs, stat].

Jing Huang, Zhengxuan Wu, Christopher Potts, Mor Geva, and Atticus Geiger. 2024. RAVEL: Evaluating interpretability methods on disentangling language model representations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8669–8687, Bangkok, Thailand. Association for Computational Linguistics.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. 2024. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*.

Aapo Hyvarinen and Hiroshi Morioka. 2016. Unsupervised Feature Extraction by Time-Contrastive Learning and Nonlinear ICA. *arXiv:1605.06336 [cs, stat]*. ArXiv: 1605.06336.

Aapo Hyvarinen and Hiroshi Morioka. 2017. Nonlinear ICA of Temporally Dependent Stationary Sources. In *Artificial Intelligence and Statistics*, pages 460–469. PMLR. ISSN: 2640-3498.

Aapo Hyvarinen, Hiroaki Sasaki, and Richard E. Turner. 2019. Nonlinear ICA Using Auxiliary Variables and Generalized Contrastive Learning. *arXiv:1805.08651 [cs, stat]*. ArXiv: 1805.08651.

Aapo Hyvärinen and Petteri Pajunen. 1999. Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, 12(3):429–439.

Shruti Joshi, Andrea Dittadi, Sébastien Lachapelle, and Dhanya Sridhar. 2025. Identifiable Steering via Sparse Autoencoding of Multi-Concept Shifts. *arXiv preprint*. ArXiv:2502.12179 [cs].

Shruti Joshi, Théo Saulus, Wieland Brendel, Philippe Brouillard, Dhanya Sridhar, and Patrik Reizinger. 2026. Who guards the guardians? The challenges of evaluating identifiability of learned representations. *Preprint*, arXiv:2602.24278.

Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. 2020a. Variational Autoencoders and Nonlinear ICA: A Unifying Framework. In *International Conference on Artificial Intelligence and Statistics*, pages 2207–2217. PMLR. ISSN: 2640-3498.

Ilyes Khemakhem, Ricardo Pio Monti, Diederik P. Kingma, and Aapo Hyvärinen. 2020b. ICE-BeeM: Identifiable Conditional Energy-Based Deep Models Based on Nonlinear ICA. *arXiv:2002.11537 [cs, stat]*. ArXiv: 2002.11537.

Sebastien Lachapelle, Tristan Deleu, Divyat Mahajan, Ioannis Mitliagkas, Yoshua Bengio, Simon Lacoste-Julien, and Quentin Bertrand. 2023a. Synergies between Disentanglement and Sparsity: Generalization and Identifiability in Multi-Task Learning. In *Proceedings of the 40th International Conference on Machine Learning*, pages 18171–18206. PMLR. ISSN: 2640-3498.

Sébastien Lachapelle, Pau Rodríguez López, Yash Sharma, Katie Everett, Rémi Le Priol, Alexandre Lacoste, and Simon Lacoste-Julien. 2021. Disentanglement via Mechanism Sparsity Regularization: A New Principle for Nonlinear ICA. *arXiv:2107.10098 [cs, stat]*. ArXiv: 2107.10098.

Sébastien Lachapelle, Divyat Mahajan, Ioannis Mitliagkas, and Simon Lacoste-Julien. 2023b. Additive Decoders for Latent Variables Identification and Cartesian-Product Extrapolation. *arXiv preprint*. ArXiv:2307.02598 [cs, stat].

Elliot Layne, Jason Hartford, Sébastien Lachapelle, Mathieu Blanchette, and Dhanya Sridhar. 2025. Sparsity regularization via tree-structured environments for disentangled representations. *Transactions on Machine Learning Research*.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, Janos Kramar, Anca Dragan, Rohin Shah, and Neel Nanda. 2024. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 278–300, Miami, Florida, US. Association for Computational Linguistics.

Phillip Lippe, Sara Magliacane, Sindy Löwe, Yuki M. Asano, Taco Cohen, and Efstratios Gavves. 2022. CITRIS: Causal Identifiability from Temporal Intervened Sequences. *arXiv preprint*. Number: arXiv:2202.03169 arXiv:2202.03169 [cs, stat].

Phillip Lippe, Sara Magliacane, Sindy Löwe, Yuki M. Asano, Taco Cohen, and Efstratios Gavves. 2023. BISCUIT: Causal Representation Learning from Binary Interactions. *arXiv preprint*. ArXiv:2306.09643 [cs, stat].

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. 2019. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations. In *International Conference on Machine Learning*, pages 4114–4124. PMLR. ISSN: 2640-3498.

Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. 2020a. Weakly-Supervised Disentanglement Without Compromises. *arXiv:2002.02886 [cs, stat]*. ArXiv: 2002.02886.

Francesco Locatello, Michael Tschannen, Stefan Bauer, Gunnar Rätsch, Bernhard Schölkopf, and Olivier Bachem. 2020b. Disentangling Factors of Variation Using Few Labels. *arXiv:1905.01258 [cs, stat]*. ArXiv: 1905.01258.

Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. 2025. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. In *The Thirteenth International Conference on Learning Representations*.

Samuel Marks and Max Tegmark. 2024. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *First Conference on Language Modeling*.

Gemma E. Moran, Dhanya Sridhar, Yixin Wang, and David M. Blei. 2022. Identifiable Deep Generative Models via Sparse Decoding. Technical Report arXiv:2110.10804, arXiv. ArXiv:2110.10804 [cs, stat] type: article.

Aaron Mueller, Jannik Brinkmann, Millicent Li, Samuel Marks, Koyena Pal, Nikhil Prakash, Can Rager, Aruna Sankaranarayanan, Arnab Sen Sharma, Jiuding Sun, Eric Todd, David Bau, and Yonatan Belinkov. 2025a. The quest for the right mediator: Surveying mechanistic interpretability for nlp through the lens of causal mediation analysis. *Computational Linguistics*, pages 1–48.

Aaron Mueller, Atticus Geiger, Sarah Wiegreffe, Dana Arad, Iván Arcuschin, Adam Belfki, Yik Siu Chan, Jaden Fried Fiotto-Kaufman, Tal Haklay, Michael Hanna, Jing Huang, Rohan Gupta, Yaniv Nikankin, Hadas Orgad, Nikhil Prakash, Anja Reusch, Aruna Sankaranarayanan, Shun Shao, Alessandro Stolfo, and 4 others. 2025b. MIB: A mechanistic interpretability benchmark. In *Forty-second International Conference on Machine Learning*.

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

Judea Pearl. 2009. Causal inference in statistics: An overview. *Statistics Surveys*, 3(none).

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. 2018. Elements of causal inference: foundations and learning algorithms. *Journal of Statistical Computation and Simulation*, 88(16):3248–3248.

Patrik Reizinger, Luigi Gresele, Jack Brady, Julius von Kügelgen, Dominik Zietlow, Bernhard Schölkopf, Georg Martius, Wieland Brendel, and Michel Besserve. 2023a. Embrace the Gap: VAEs Perform Independent Mechanism Analysis. *arXiv preprint*. ArXiv:2206.02416 [cs, stat].

Patrik Reizinger, Siyuan Guo, Ferenc Huszár, Bernhard Schölkopf, and Wieland Brendel. 2024. Identifiable Exchangeable Mechanisms for Causal Structure and Representation Learning.

Patrik Reizinger, Yash Sharma, Matthias Bethge, Bernhard Schölkopf, Ferenc Huszár, and Wieland Brendel. 2023b. Jacobian-based Causal Discovery with Nonlinear ICA. *Transactions on Machine Learning Research*.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. 2021. Towards Causal Representation Learning. *arXiv:2102.11107 [cs]*. ArXiv: 2102.11107 version: 1.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. *Preprint*, arXiv:1312.6034.

Nishant Subramani, Nivedita Suresh, and Matthew Peters. 2022. Extracting latent steering vectors from pretrained language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 566–581, Dublin, Ireland. Association for Computational Linguistics.

Raphael Suter, Djordje Miladinovic, Bernhard Schölkopf, and Stefan Bauer. 2019. Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. In *International Conference on Machine Learning*, pages 6056–6065. PMLR.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, and 89 others. 2024. Gemma: Open models based on gemini research and technology. *Preprint*, arXiv:2403.08295.

Julius von Kügelgen. 2024. Identifiable Causal Representation Learning: Unsupervised, Multi-View, and Multi-Environment.

Julius von Kügelgen, Michel Besserve, Liang Wendong, Luigi Gresele, Armin Kekić, Elias Bareinboim, David M. Blei, and Bernhard Schölkopf. 2023. Nonparametric Identifiability of Causal Representations from Unknown Interventions. *arXiv preprint*. ArXiv:2306.00542 [cs, stat].

Julius von Kügelgen, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. 2021. Self-Supervised Learning with Data Augmentations Provably Isolates Content from Style. ArXiv: 2106.04619.

Liang Wendong, Armin Kekić, Julius von Kügelgen, Simon Buchholz, Michel Besserve, Luigi Gresele, and Bernhard Schölkopf. 2023. Causal Component Analysis. *arXiv preprint*. ArXiv:2305.17225 [cs, stat].

Thaddäus Wiedemer, Prasanna Mayilvahanan, Matthias Bethge, and Wieland Brendel. 2023. Compositional Generalization from First Principles. *arXiv preprint*. ArXiv:2307.05596 [cs, stat].

Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D Manning, and Christopher Potts. 2025. AxBench: Steering LLMs? even simple baselines outperform sparse

autoencoders. In *Forty-second International Conference on Machine Learning*.

Jiaqi Zhang, Chandler Squires, Kristjan Greenewald, Akash Srivastava, Karthikeyan Shanmugam, and Caroline Uhler. 2023. Identifiability Guarantees for Causal Disentanglement from Soft Interventions. *arXiv preprint*. ArXiv:2307.06250 [cs, math, stat].

Zuheng, Xu, Moksh Jain, Ali Denton, Shawn Whitfield, Aniket Didolkar, Berton Earnshaw, and Jason Hartford. 2024. Automated discovery of pairwise interactions from unstructured data. *Preprint*, arXiv:2409.07594.

A Data Generation

We use probabilistic context-free grammars (PCFGs) to generate the training data for our SAEs. Non-terminals have attributes corresponding to the ground-truth concepts. In Figure 6, we show a subsample of the rules in the grammar. Note that this sample is simplified: most terminal-generating rules have over 10 non-terminals, and there are more sentence templates than displayed in the figure.

Our PCFG has 7 templates per tense/voice combination. We define non-terminals over combinations of concepts. There are 9 subject and 9 object non-terminals per domain/sentiment combination ($4 \cdot 3 \cdot 9 = 108$ subjects and objects each). We have 12 verbs per tense/sentiment/voice combination ($2 \cdot 3 \cdot 2 \cdot 12 = 144$ total), which can be freely combined to yield $144 \cdot 108 \cdot 108 = 1{,}679{,}616$ sentences. Optional prepositional phrases (8 per sentiment), adverbs (10 per sentiment), and compound sentences push the total to over 1 billion possible sentences. For compound sentences, both clauses share identical concept labels.

Concepts are uniformly distributed by default, with approximately $\frac{T}{V}$ examples per concept value, where V is the number of values per concept, and T is the dataset size. When there are cross-concept correlations, the correlated concepts are upsampled. For example, correlating domain=science with sentiment=positive means sentiment is sampled uniformly for non-science domains, but sentiment=positive is upsampled when domain=science.

In Table 3, we show examples from our generated training set. When we generate without correlations between concepts, there is an approximately uniform distribution of each concept, and correlations of approximately 0 across all concept pairs. If a concept-value pair is correlated, we pre-compute the example set such that we can achieve the closest match to the desired correlation. When training SAEs, we iterate for multiple epochs over the full dataset (when there are no cross-concept correlations) or the subsampled dataset (when there are cross-concept correlations).

B SAE Training Details

B.1 SAE Architectures

Here, we define sparse autoencoders and describe the differences between the architectures we study.

Sparse autoencoders. The conceptually simplest architecture we deploy is the ReLU sparse autoencoder (Huben et al., 2024; Bricken et al., 2023), which learns a mapping from $\mathbf{x} = \mathbf{h}^{\ell}$ to a learned sparse feature vector $\mathbf{f}$, and then reconstructs the activations $\hat{\mathbf{x}}$ given $\mathbf{f}$. More formally:

$$\mathbf{f} = \text{ReLU}(W_{\text{enc}}\mathbf{x} + \mathbf{b}_{\text{enc}}) \tag{7}$$

$$\hat{\mathbf{x}} = W_{\text{dec}}(\mathbf{f} - \mathbf{b}_{\text{enc}}) + \mathbf{b}_{\text{dec}} \tag{8}$$

ReLU SAEs minimize $\mathcal{L} = \text{MSE}(\mathbf{x}, \hat{\mathbf{x}}) + \lambda\|\mathbf{f}\|_1$.

Top-K SAEs (Gao et al., 2025) are similar to ReLU SAEs, but they strictly retain the top k activations per sample and zero out all others:

$$\mathbf{f} = \text{top-}k(W_{\text{enc}}\mathbf{x} + \mathbf{b}_{\text{enc}}) \tag{9}$$

Sparsemax distance encoders (SpADE) can capture nonlinearly separable and heterogeneous features; we refer readers to Hindupur et al. (2025) for details. In formal terms:

$$\mathbf{f} = \text{Sparsemax}(-\lambda d(\mathbf{x}, W)) \tag{10}$$

where $d(\mathbf{x}, W)_i = \|\mathbf{x} - W_i\|_2^2$. Hindupur et al. (2025) show that this architecture can capture more irregular concept geometries, whereas ReLU SAEs assume linear separability, and Top-K SAEs assume angular separability.

Sparse shift autoencoders. Sparse shift autoencoders (SSAEs; Joshi et al., 2025) are trained using paired observations $(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ assumed to be sampled from the following generative process:

$$S \sim p(S), \quad (\boldsymbol{c}, \tilde{\boldsymbol{c}}) \sim p(\boldsymbol{c}, \tilde{\boldsymbol{c}} \mid S), \tag{11}$$

$$\boldsymbol{x} := g(\boldsymbol{c}), \quad \tilde{\boldsymbol{x}} := g(\tilde{\boldsymbol{c}})\,, \tag{12}$$

where $S \subseteq \{1, \ldots, d_c\}$ denotes the subset of concepts that vary between $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$, and d_c represents the dimension of *varying concepts*, the concepts that are intervened upon in the dataset.

Note that SSAEs take as input *difference vectors* $\Delta\mathbf{z} := f(\tilde{\boldsymbol{x}}) - f(\boldsymbol{x}) = \tilde{\boldsymbol{z}} - \boldsymbol{z}$ that represent concept differences in activation space and model them as:

$$\Delta\hat{\mathbf{c}}_V := r(\Delta\mathbf{z}) := \mathbf{W}_e(\Delta\mathbf{z} - \mathbf{b}_d) + \mathbf{b}_e\,; \tag{13}$$

$$\Delta\hat{\mathbf{z}} := q(\Delta\hat{\mathbf{c}}_V) := \mathbf{W}_d\Delta\hat{\mathbf{c}}_V + \mathbf{b}_d \tag{14}$$

where $r : \mathbb{R}^{d_z} \to \mathbb{R}^{d_c}$ is an affine encoder $q : \mathbb{R}^{d_c} \to \mathbb{R}^{d_z}$ is an affine decoder. In words, the representation $r(\Delta\mathbf{z})$ predicts $\Delta\mathbf{c}_V$, i.e., the concept shifts corresponding to $\Delta\mathbf{z}$.

```
S[active, present] → Subj V O | Subj V O PP | Adv, Subj V O
S[passive, past] → Subj V_past O | Subj V_past O PP | Adv, Subj V_past O
S[passive, present] → O is V_pp by Subj | Adv, O is V_pp
                        | O is being V_pp by Subj
S[passive, past] → O was V_pp by Subj | Adv, O was V_pp
                     | O had been V_pp by Subj

Subj[news, positive] → the successful team | the innovative company
Subj[news, neutral] → the government | the company
Subj[fantasy, negative] → the evil sorcerer | the treacherous assassin
V[negative] → criticizes | condemns | rejects
V[neutral] → announces | reports | explains
V_past[positive] → celebrated | praised | endorsed

PP[neutral] → in recent days | across different sectors
PP[positive] → with remarkable success | beyond expectations
PP[negative] → without proper justification | to widespread criticism
```

Figure 6: Excerpts from the context-free grammar we use to generate the SAE training and evaluation datasets.

Concept Label				
Voice	Tense	Domain	Sentiment	**Example Sentence**
Active	Present	Science	Positive	The brilliant scientist celebrates the remarkable findings.
Active	Present	Science	Neutral	The expert announces the parameters in recent days.
Active	Present	Science	Negative	As of today, the discredited theory rejects the inconclusive evidence.
Active	Past	Fantasy	Negative	Unsuccessfully, the malevolent dragon damaged the corrupted land.
Passive	Past	News	Neutral	The event was explained in the recent report.
Passive	Present	Other	Positive	The pleasant surprise is endorsed advantageously by the talented artist.
Passive	Past	Other	Neutral	The question was answered when the family announced the event.

Table 3: Examples of sentences generated by our PCFG.

SSAEs are trained to solve the following constrained problem:

$$(\hat{r}, \hat{q}) \in \arg\min_{r,q} \mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{x}}} \left[||\Delta \mathbf{z} - q(r(\Delta \mathbf{z}))||_2^2 \right] \quad (15)$$

$$\text{s.t. } \mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{x}}} ||r(\Delta \mathbf{z})||_0 \leq \beta \,, \quad (16)$$

where Eq. 15 is the standard auto-encoding loss that encourages good reconstruction and Eq. 16 is a regularizer that encourages the predicted concept shift vector $\Delta \hat{\mathbf{c}}_V \coloneqq \hat{r}(\Delta \mathbf{z})$ to be sparse. Since the ℓ_0-norm is non-differentiable, in practice we replace it by an ℓ_1-norm leading to the following relaxed sparsity constraint:

$$\mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{x}}} ||r(\Delta \mathbf{z})||_1 \leq \beta \,. \quad (17)$$

We then approximately solve this constrained problem by finding a saddle point of its Lagrangian using the ExtraAdam algorithm (Gidel et al., 2020) as implemented by Gallego-Posada and Ramirez (2022).

B.2 Hyperparameters

Sparse autoencoders. All Pythia-70M sparse autoencoders are trained using a batch size of 128 sequences for 10000 steps. We train on the output of the middle layer (layer 3). Optimization is performed using Adam with an initial learning rate of 1×10^{-3}, 200 warmup steps, and $\beta_1 = 0.9, \beta_2 = 0.95$. Top-$k$ SAEs are trained with $k = 128$. For Gemma-2-2B, we use the same hyperparameters for all SAEs except SpADE, which has higher memory requirements; for this architecture, we reduce the batch size to 64 while maintaining all other hyperparameters.[9] We also train on the middle layer (layer 13). Our implementation is based on that of Hindupur et al. (2025).

Sparse shift autoencoders. For SSAEs, we must train on pairwise differences in activations. For this, we iterate over the training set to get example x_i, and then uniformly sample another example x_j, ensuring that $i \neq j$. Otherwise, we use similar hyperparameters as when training SAEs. Note that SSAEs should be trained on the *final* layer of a model, rather than the

[9] We experimented with doubling the number of training steps to compensate for the halved batch size for Gemma-2-2B SpADE SAEs. Final loss reductions were very small, so we chose to continue using 10000 iterations for uniformity.

SAE Arch.	$\rho(z_i, z_j)$	NMSE	Var. Explained	% Sparsity
ReLU	0.0	0.004 (0.000)	99.6 (0.0)	58.7 (0.2)
	0.1	0.006 (0.000)	99.7 (0.0)	54.6 (0.1)
	0.2	0.006 (0.008)	99.7 (0.0)	54.4 (0.2)
	0.5	0.007 (0.000)	99.7 (0.0)	54.4 (0.1)
	0.9	0.003 (0.000)	99.7 (0.0)	54.4 (0.1)
	1.0	0.003 (0.000)	99.7 (0.1)	54.4 (0.1)
Top-K	0.0	0.058 (0.000)	94.2 (0.0)	97.6 (0.0)
	0.1	0.056 (0.000)	94.4 (0.0)	97.6 (0.0)
	0.2	0.056 (0.000)	94.4 (0.0)	97.6 (0.0)
	0.5	0.060 (0.000)	94.0 (0.0)	97.6 (0.0)
	0.9	0.057 (0.000)	94.4 (0.0)	97.6 (0.0)
	1.0	0.064 (0.000)	93.6 (0.0)	97.6 (0.0)
SpADE	0.0	0.003 (0.000)	99.7 (0.0)	58.8 (0.3)
	0.1	0.003 (0.000)	99.7 (0.0)	58.8 (0.2)
	0.2	0.003 (0.000)	99.7 (0.0)	59.3 (0.1)
	0.5	0.003 (0.000)	99.7 (0.0)	60.6 (0.0)
	0.9	0.004 (0.000)	99.6 (0.0)	64.9 (0.1)
	1.0	0.005 (0.000)	99.5 (0.0)	66.8 (0.1)
Natural	-	0.005	99.5	99.8
SSAE	0.0	0.004 (0.001)	99.6	98.8 (0.0)
	0.1	0.004 (0.001)	99.6 (0.0)	99.1 (0.0)
	0.2	0.005 (0.001)	99.6 (0.0)	99.0 (0.0)
	0.5	0.005 (0.001)	99.6 (0.0)	99.1 (0.0)
	0.9	0.005 (0.002)	99.6 (0.0)	99.0 (0.0)
	1.0	0.004 (0.001)	99.6 (0.0)	99.2 (0.0)

Table 4: Variance explained, losses, and sparsities for SAEs trained on the middle layer of Pythia-70M (or last layer in the case of SSAEs). SSAE results are not comparable to those of other SAEs; unlike other architectures, they are trained and evaluated on *pairwise differences* of activations.

SAE Arch.	$\rho(z_i, z_j)$	NMSE	Var. Explained	% Sparsity
ReLU	0.0	0.014 (0.000)	98.7 (0.0)	49.6 (0.1)
	0.1	0.014 (0.000)	98.6 (0.0)	49.6 (0.1)
	0.2	0.014 (0.000)	98.7 (0.0)	49.6 (0.0)
	0.5	0.014 (0.000)	98.7 (0.0)	49.9 (0.0)
	0.9	0.011 (0.000)	98.9 (0.0)	50.0 (0.0)
	1.0	0.010 (0.000)	99.0 (0.0)	50.0 (0.0)
Top-K	0.0	0.218 (0.001)	78.1 (0.001)	99.4 (0.0)
	0.1	0.218 (0.000)	78.1 (0.000)	99.4 (0.0)
	0.2	0.216 (0.001)	78.3 (0.000)	99.4 (0.0)
	0.5	0.218 (0.000)	78.2 (0.000)	99.4 (0.0)
	0.9	0.236 (0.000)	76.4 (0.000)	99.4 (0.0)
	1.0	0.269 (0.000)	73.1 (0.000)	99.4 (0.0)
SpADE	0.0	0.094 (0.0)	90.6 (0.0)	96.9 (0.1)
	0.1	0.091 (0.000)	90.5 (0.0)	96.8 (0.0)
	0.2	0.091 (0.001)	90.4 (0.0)	96.9 (0.0)
	0.5	0.099 (0.001)	89.5 (0.0)	96.7 (0.0)
	0.9	0.149 (0.000)	84.4 (0.1)	96.9 (0.0)
	1.0	0.167 (0.001)	84.5 (0.1)	96.2 (0.0)
Natural	-	0.064	93.6	99.6
SSAE	0.0	0.064 (0.001)	98.8 (0.0)	91.9 (0.1)
	0.1	0.068 (0.000)	98.8 (0.0)	91.5 (0.0)
	0.2	0.061 (0.000)	98.8 (0.0)	91.6 (0.1)
	0.5	0.072 (0.000)	98.8 (0.0)	91.4 (0.0)
	0.9	0.069 (0.000)	98.9 (0.0)	91.3 (0.0)
	1.0	0.074 (0.001)	99.0 (0.0)	91.4 (0.0)

Table 5: Variance explained, losses, and sparsities for SAEs trained on the middle layer of Gemma-2-2B (or last layer in the case of SSAEs). SSAE results are not comparable to those of other SAEs; unlike other architectures, they are trained and evaluated on *pairwise differences* of activations.

middle layer: this choice is motivated by the claim that concepts in the output space are most easily linearly identified in the final layer (Joshi et al., 2025).[10]

We present NMSE, variance explained, and percent sparsity on the test set in Table 4 (for Pythia) and Table 5 (for Gemma).

Probes. All probes are logistic regression probes. The probes used in correlational experiments are trained on the middle layer of Pythia-70M or Gemma-2-2B for a maximum of 1000 steps. We use the implementation of `scikit-learn` (Pedregosa et al., 2011).[11] k-sparse probes are identical in architecture and hyperparameters, but we filter the set of neurons or features to reduce dimensionality before training the probes (and also train them on featurized representations rather than the original activation space); see App. G for details. For the binary probes, we balance the training dataset of each probe by uniformly subsampling the more frequent class such that the number of examples for both classes is the same.

For the multinomial probes used for evaluating steering, the architecture and hyperparameters are the same, except that the probe outputs one logit *per concept value* rather than a single logit. These probes are trained on the final layer of Pythia-70M or Gemma-2-2B, as their purpose is to estimate the probability of a concept appearing in the model's output. Note that we do not rebalance the data for multinomial probes; we only train multinomial probes on data where there are no cross-concept correlations, so there is already an approximately uniform distribution of labels for each probe's training set.

[10] Using different layers for different SAE architectures introduces a confound. However, in pilot experiments, we found that other architectures tended to yield worse disentanglement and steering results when trained on the final layer. Thus, the current locations seem to be closer to optimal than training all SAEs on the same layer.

[11] Specifically, we use the Newton-Cholesky solver.

C Identifiability Definitions

Identifiability definitions formulate the permissible transformations—termed an equivalence class—of the learned latent factors $\mathbf{f}$ by such that the resulting probability distributions parametrized by the neural network are equivalent. The smaller the equivalence class, the stronger assumptions are generally required.

Definition 1 (Strong Identifiability (Khemakhem et al., 2020b))**.** *Given a parameter class Θ, when the feature extractors $\mathcal{F}_{\theta_1}, \mathcal{F}_{\theta_2}$ produce latent representations $\mathbf{f}_1 = \mathcal{F}_{\theta_1}(\mathbf{x}), \mathbf{f}_2 = \mathcal{F}_{\theta_2}(\mathbf{x})$ from observations $\mathbf{x}$ that are equivalent up to scaled permutations and offsets c for all $\theta_1, \theta_2 \in \Theta$, i.e.,*

$$\theta_1 \sim \theta_2 \iff \mathbf{f} = \mathcal{F}_{\theta_1}(\mathbf{x}) = \mathbf{D}\mathbf{P}\mathcal{F}_{\theta_2}(\mathbf{x}) + c, \quad (18)$$

where $\mathbf{D}$ is a diagonal and $\mathbf{P}$ a permutation matrix. Then θ_1, θ_2 fulfill an equivalence relationship.

Definition 2 (Weak Identifiability (Khemakhem et al., 2020b)). *Given a parameter class Θ, when the feature extractors $\mathcal{F}_{\theta_1}, \mathcal{F}_{\theta_2}$ produce latent representations $\mathbf{f}_1 = \mathcal{F}_{\theta_1}(\mathbf{x}), \mathbf{f}_2 = \mathcal{F}_{\theta_2}(\mathbf{x})$ from observations $\mathbf{x}$ that are equivalent up to matrix multiplications and offsets c for all $\theta_1, \theta_2 \in \Theta$, i.e.,*

$$\theta_1 \sim \theta_2 \iff \mathbf{f} = \mathcal{F}_{\theta_1}(\mathbf{x}) = \mathbf{A}\mathcal{F}_{\theta_2}(\mathbf{x}) + c, \quad (19)$$

where $\operatorname{rank}(\mathbf{A}) \geq \min(\dim \mathbf{f}; \dim \mathcal{X})$. Then θ_1, θ_2 fulfill an equivalence relationship.

Definition 3 (Identifiability up to elementwise nonlinearities (Hyvarinen and Morioka, 2017)). *Given a parameter class Θ, when the feature extractors $\mathcal{F}_{\theta_1}, \mathcal{F}_{\theta_2}$ produce latent representations $\mathbf{f}_1 = \mathcal{F}_{\theta_1}(\mathbf{x}), \mathbf{f}_2 = \mathcal{F}_{\theta_2}(\mathbf{x})$ from observations $\mathbf{x}$ that are equivalent up to elementwise nonlinearities, matrix multiplications and offsets c for all $\theta_1, \theta_2 \in \Theta$, i.e.,*

$$\theta_1 \sim \theta_2 \iff \mathbf{f} = \mathcal{F}_{\theta_1}(\mathbf{x}) = \mathbf{A}\sigma\left[\mathcal{F}_{\theta_2}(\mathbf{x})\right] + c, \quad (20)$$

where $\operatorname{rank}(\mathbf{A}) \geq \min(\dim \mathbf{f}; \dim \mathcal{X})$ and σ denotes an elementwise nonlinear transformation. Then θ_1, θ_2 fulfill an equivalence relationship.

D Metrics

D.1 MCC

Given a set of ground-truth concepts $\{z_1, \ldots, z_n\}$ that generate an input example $\mathbf{x}$ where each concept $z_j \in Z$, then $\forall i \in [1, \ldots, n]$, we compute $\hat{\mathbf{f}}_j = \arg\max_i |\rho_{\mathcal{D}}(f_i, z_j)|$, where f_i is the activation of feature $\mathbf{f}_i$ and ρ is the Pearson correlation. Intuitively, $\hat{\mathbf{f}}_j$ is the feature whose activation correlates most with the value of z_j on some training dataset $\mathcal{D}$. Given test set $\mathcal{T}$ where concepts are uniformly distributed w.r.t. each other (i.e., no built-in correlations), we use $\rho_{\mathcal{T}}(\hat{\mathbf{f}}_j, z_j)$ as a measure of how well the featurizer linearly identifies concept z_j. After locating the best features $\{\hat{\mathbf{f}}_j\}_{j=1}^n$ for each concept, we compute the MCC as the mean of their correlations with their respective concepts on $\mathcal{T}$. In other words:

$$\text{MCC} = \frac{1}{n}\sum_{j=1}^{n} \rho_{\mathcal{T}}(\hat{f}_j, z_j). \quad (21)$$

The MCC is measured using one-dimensional features, but multinomial concepts may not be one-dimensional in $\mathbf{f}$ or $\mathbf{h}^\ell$ (Engels et al., 2025). Thus, to create a fairer evaluation, we compute the MCC over binarized concepts. That is, given a variable $z_i \in Z$ with V_i possible values, we create a new binary variable $v_{i,x} \in \mathbb{B}$ for each value x corresponding to whether $z_i = v_{i,x}$. When computing the MCC, we first average the correlation coefficients for all $v_{i,x} \in V_i$ before taking the macroaverage across concepts.

D.2 DCI-ES

Here, we provide further detail on the DCI-ES metrics (Eastwood et al., 2023), and give methodological details as to how we compute them. Our implementation is based directly on that of Eastwood et al. (2023).

DCI-ES stands for **d**isentanglement, **c**ompleteness, **i**nformativeness, **e**xplicitness, and **s**ize. We focus on the first four metrics, as these are the most relevant to establishing identifiability. Disentanglement and completeness require us to first compute importance matrix $M \in \mathbb{R}^{|\mathbf{f}| \times |Z|}$. For example, if we train a multinomial probe to predict concept z_j from feature $\mathbf{f}_i$, we can compute the importance of each dimension of $\mathbf{f}$ post hoc. Each concept z_j defines a column of M. Note that $\forall i, j : M_{ij} \geq 0$, and $\sum_{i=1}^{|\mathbf{f}|} M_{ij} = 1$.

Disentanglement measures the average number of concepts z_j that are captured by any single feature $\mathbf{f}_i$. To compute it, we first compute the entropy $H_Z(P_{i\cdot})$ of the distribution $P_{i\cdot}$ defined over row i of M: $P_{ij} = \frac{M_{ij}}{\sum_k M_{ik}}$. Disentanglement is then defined as $D_i = 1 - H_K(P_{i\cdot})$. This score is maximized when feature $\mathbf{f}_i$ is only responsible for predicting a single concept z_j; it is minimized when feature $\mathbf{f}_i$ is equally important for predicting all concepts.

Completeness measures the average number of features $\mathbf{f}_i$ that are useful in predicting a single concept z_j. This score is defined analogously to disentanglement, but over columns j in M: we take $C_j = 1 - H_{\mathbf{f}}(P_{\cdot j})$. Completeness is maximized when only one feature $\mathbf{f}_i$ is helpful in predicting z_j, and it is minimized when all features are equally important in predicting the concept.

Informativeness is inversely proportional to the prediction error of a probe trained on the feature vector. In the implementation of Eastwood et al. (2023), it is simply defined as the accuracy of a probe in predicting concept z_j when trained on the feature vector $\mathbf{f}$. This captures whether a ground-truth concept is recoverable from the feature vector.

Explicitness is conceptually related to informativeness. E captures the trade-off between the probe's capacity and the probe loss; this is measured as one minus the normalized area under the loss-capacity curve (AULCC); we refer readers to Eastwood et al. (2023) for details. This score is maximized when the lowest-capacity probe achieves the best loss, and thus that no excess capacity was required to fully recover a given concept.

D.3 Further Details on Independence and Disjointness

To illustrate the conceptual distinction between independence and disjointness, we present diagrams in Figure 7. Intuitively, disjointness implies that the influence of one feature on the output does not interact with another, and thus that the effect of steering both features can be predicted from the result of steering either in isolation. Independence instead implies that steering with one concept would not affect how the model uses other concepts. Refer to §5.2 for details.

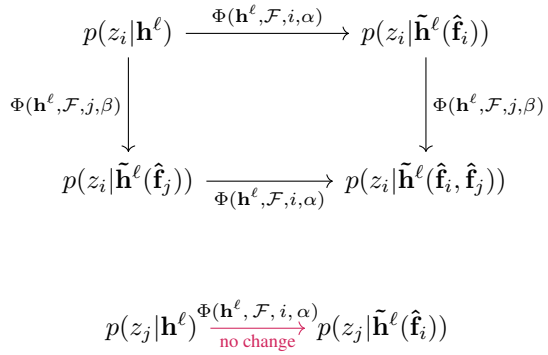

Figure 7: **The difference between disjointness and independence: (Top)** Two concepts z_i and z_j with feature representations $\hat{\mathbf{f}}_i$ and $\hat{\mathbf{f}}_j$, respectively, are disjoint if the top diagram commutes. **(Bottom)** If they are independent then there is no commutative relationship, as steering with $\hat{\mathbf{f}}_i$ should not affect $p(z_j)$. Intuitively, disjointness implies that two features have no interaction terms, and thus that the effect of steering of both can be predicted from the result of steering either in isolation. Feature selectivity and concept independence instead imply that steering with one concept's top feature does not affect how the model uses other concepts. Refer to §5.2 for formulae and empirical details.

E Is One Dimension Sufficient?

In SAE-based interpretability studies, it is common to steer with a single feature, regardless of how many features receive high attributions for a given task. This corresponds to the following assumption:

Assumption: *Given binary concept z_i and feature vector $\mathbf{f}$, one dimension $\mathbf{f}_i$ of $\mathbf{f}$ is sufficient to represent and control z_i in $\mathcal{M}$.*

To evaluate the extent to which this assumption holds in practice, we train k-sparse probes (as operationalized in Gurnee et al. (2023)) on featurized representations $\mathbf{f}$. k-sparse probes are linear probes that may have non-zero weights from up to k dimensions of the representations they are trained on. Lachapelle et al. (2023a) established a connection between disentanglement and sparse prediction: they prove that disentanglement leads to optimal loss using sparse predictors. Further, as features become more entangled, we need to reduce sparsity regularization to maintain accuracy; this theoretical finding further motivates the following experiment.

Hypothesis. More dimensions yield monotonically increasing expressive power. Thus, performance should be non-decreasing as k increases. We care primarily about when increasing k begins to yield diminishing improvements in the MCC. Representations obtained with strong sparsity constraints, like SAEs, should reach this saturation point at smaller k than representations with no such constraints, such as residual vectors.

Results. The MCCs of k-sparse probes trained on feature vectors $\mathbf{f}$ are presented in Figure 8. Top-K SAEs and SSAEs achieve the best trade-off between MCC and sparsity at all k; they also approach the MCC of training a normal probe on the full activation vector at the residual stream. ReLU SAEs do not begin saturating even at 10–50 features, whereas all other SAEs do. SSAEs and Top-K SAEs achieve better concept recovery at the same k as the residual neuron baseline, whereas ReLU SAEs do not.

These results suggest that SAEs do confer sparsity benefits compared to the original activation space of $\mathcal{M}$, but also that one-dimensionality assumptions may often yield suboptimal results—even when the concepts are relatively simple. That said, the relative ordering of MCCs across architectures does not generally change with k, so comparisons across architectures should generally yield conclusions that are stable across choices of k (at least up to $k = 50$).

F Probe Accuracies

Here, we present the accuracies of each probe we use in our disentanglement experiments and evaluations. We present these as heatmaps to verify whether each probe learns an independent representation of its target concept; if it does, we expect high scores along the diagonal, lower-than-random scores for within-concept pairs,[12] and random-chance scores for across-concept pairs.

Binary linear probes trained on the middle layers of Pythia-70M and Gemma-2-2B (Figure 9) achieve near-perfect accuracies on their respective concepts, and achieve the expected random accuracies on all other concepts. This empirically supports Assumption 1, and supports the idea that the MCC ceiling should be high (§4.1).

In §5.1 and §5.2, we instead use multinomial linear probes trained on the final layers of Pythia-70M and Gemma-2-2B. We find (Figure 10) that these probes also achieve the expected high accuracies on the target concepts, below-random-chance accuracies on within-concept pairs, and random-chance accuracies on across-concept pairs. This validates that the non-independence we observe in our steering experiments are not due to the probes, but rather are more likely due to the featurization methods that we use to steer.

G Sparse Probing

Here, we replicate the setup of Gurnee et al. (2023) in our cross-concept correlation setting. We aim to assess which k-sparse probing methods are more robust to cross-concept correlations at multiple k. We focus on the two most performant methods from Gurnee et al. (2023): max mean difference (MD), and logistic regression (LR). MD works by computing the average

[12] We expect lower-than-random scores for within-concept pairs because a classifier trained on an alternative value of a concept should be strictly worse than a random probe, as the target label will be *negatively* correlated with the target concept.

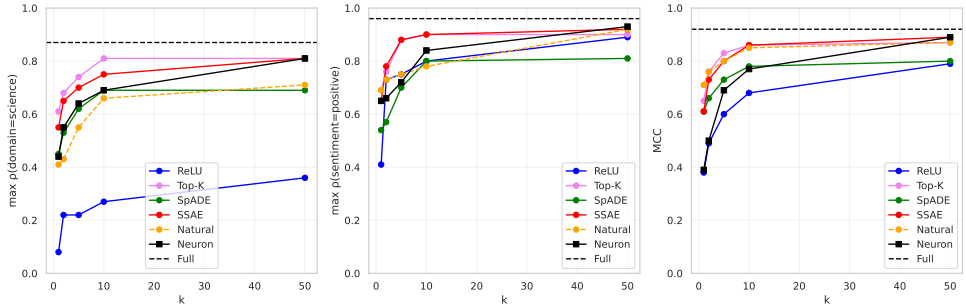

Figure 8: **Correlation coefficients between probe logits and concept labels for domain=science (left), sentiment=positive (middle), and MCC (right)**. Results for Gemma-2-2B shown here; results for Pythia-70M are in App. H. We vary the number of dimensions k that the probe is allowed to have non-zero weights from. k-sparse probes trained on SAEs begin to converge around 10 dimensions for Top-K, SpADE, SSAE, and Natural, and recover most of the performance of a non-sparse probe that is allowed to use the entire residual vector (Full). k-sparse probes trained on the residual stream (Neuron) require more dimensions to converge, as expected.

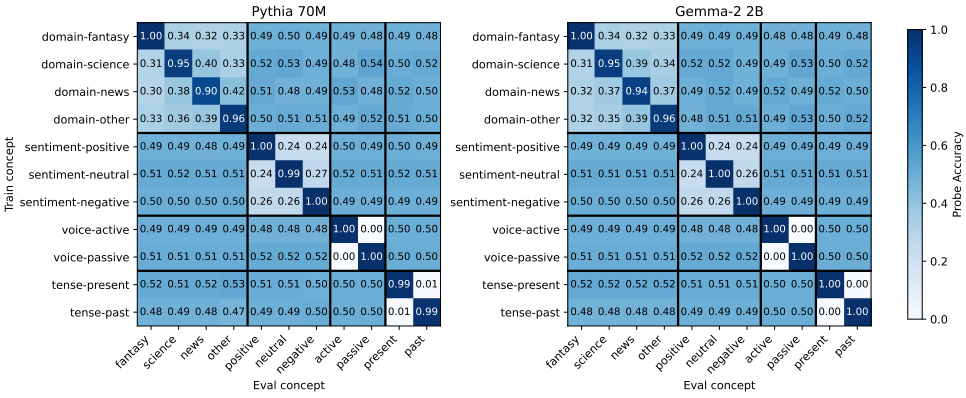

Figure 9: Accuracy of binary probes (rows) on all concept value classification tasks (columns). We expect high values on the diagonals, below random chance for within-concept value pairs, and random chance for across-concept value pairs.

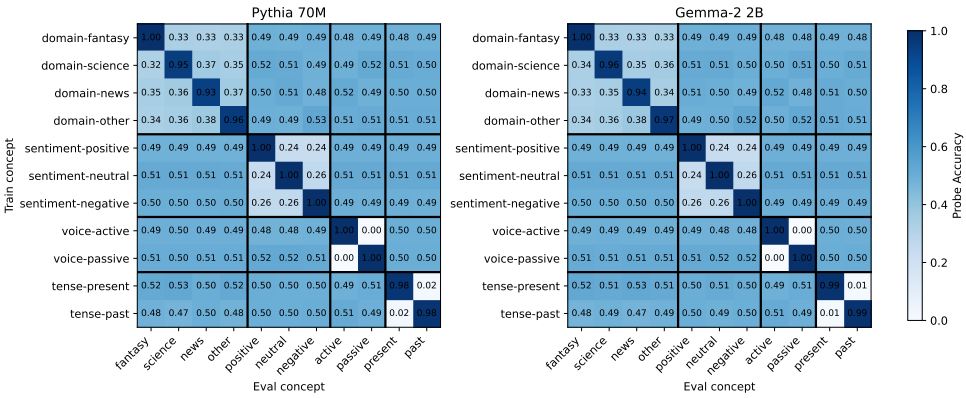

Figure 10: Accuracy of multinomial probes on all concept value classification tasks (columns). We expect high values on the diagonals, below random chance for within-concept value pairs, and random chance for across-concept value pairs.

difference in activations between positive and negative samples, and taking the k neurons whose mean activation difference is greatest. LR works by first training a logistic regression probe with L_1 regularization on the full activation vector, and then taking the top k according to the weights of the probe.

We observe (Figure 11) that the logistic regression (LR) method of selecting neurons is more effective at lower k. Between $k = 5$ and $k = 10$, MD generally overtakes LR in performance. As we are more concerned with low-dimensional concept recovery, we focus on LR in the feature dimensionality experiment (§E).

H Further Disentanglement Results

Here, we present correlation coefficients and MCCs for k-sparse probes trained with varying k on SAEs for Pythia-70M (Figure 12). As with Gemma-2-2B, correlation coefficients tend to converge at around 10 dimensions; this suggests that the one-dimensionality assumption may not often hold in practice, even for much smaller models. Note also that the neuron baseline is far more performant for Pythia than Gemma; perhaps this is because $k = 10$ represents a far greater proportion of the dimensions of $\mathbf{h}^\ell$ for Pythia than Gemma. Other trends are largely consistent with Figure 8.

I Further Steering Results

I.1 Results for Gemma 2

Here, we present steering heatmaps for Gemma-2-2B (Figure 13). Features appear less independent than for Pythia-70M, as indicated by more significant across-concept ΔLogOdds for many concept pairs. That said, the expected diagonal trend is still present. This is further evidence that SAE features do not often correspond to causally independent concept representations.

I.2 Locating Top Features with Gradient Attributions

In §5, we locate features to steer by correlating feature activations with concept labels. However, Arad et al. (2025) has found that the features that *detect* the input concept (the top-correlated features in our case) and the features that *control* the concept in a model's outputs are nearly disjoint. Thus, for steering experiments, we use gradient attributions (Simonyan et al., 2014) to locate the feature that should be steered. We use the method of Marks et al. (2025) (fold the SAE into the forward pass, backpropagate from a probe's logit) to compute gradient attributions to sparse features: given binary probe Π trained on the final layer L of a model to predict z_i, we backpropagate from $\Pi(\mathbf{h}^L)$, the probe logit, to obtain its gradient with respect to a feature activation $\frac{\partial \Pi(\mathbf{h}^L)}{\partial f_i}$. We then multiply each feature's gradient by its activation to obtain the gradient attribution $\frac{\partial \Pi(\mathbf{h}^L)}{\partial f_i} \cdot f_i$.[13] We take the feature with the maximum average attribution across examples.

Results for Pythia-70M (Figure 14) and Gemma-2 (Figure 15) show similar trends. The magnitude of ΔLogOdds is generally much higher when locating features using gradient attributions, despite the use of similar steering coefficients. This is likely because gradient attributions directly select for features with the greatest effect on the probe logits. We also observe slightly lower feature selectivities and concept independences in general; this is likely because gradient attributions directly select for large effects on the probe logits, rather than exclusion of unrelated concepts from the feature.

We also rerun the disjointness evaluation of §5.2 when selecting features using gradient attributions instead of correlations. We observe (Figure 16) that trends are largely the same. Again, the magnitude of ΔLogOdds is higher as compared to when selecting features using correlations.

J Additional Variable Correlation Experiments

The experiments thus far have focused on correlations specifically between the science domain and positive sentiment. To assess how well these results generalize to new variable correlations, we rerun the correlational experiments while instead correlating the past tense with the passive voice.

We present MCC results (Figure 17). Findings are largely consistent with Figure 2: supervised featurizers like probes perform best by far, but their performance drops sharply from ρ=0.9. Top-K SAEs and SSAEs are still among the best-performing methods among unsupervised featurizers given our data. One difference is that SAEs trained on large-scale natural language corpora are far better at recovering tense=past than our SAEs—especially for Gemma-2-2B.

K Qualitative Examples of Steering

Quantitative results suggest that steering particular features can affect how models use a concept in its outputs. Here, we sanity-check this finding by showing examples of model generations before and after steering the top feature for "domain=science" or "sentiment=positive" (Figure 18). We select the top features using the same method as in §5.1. We observe that steering generally has the expected impact on model behavior, especially when SAEs are trained on uniform distributions of concepts ($\rho = 0$). We also observe that steering with SAEs trained with complete correlations between concepts ($\rho = 1$) tend to generate outputs that have changed w.r.t. multiple concepts.

L Use of AI Tools

We used AI tools including Claude and ChatGPT for minor assistance in editing the text of this manuscript.

[13] Intuitively, this is a first-order Taylor approximation of the effect of changing feature activation f_i to 0 on $\Pi(\mathbf{h}^L)$.

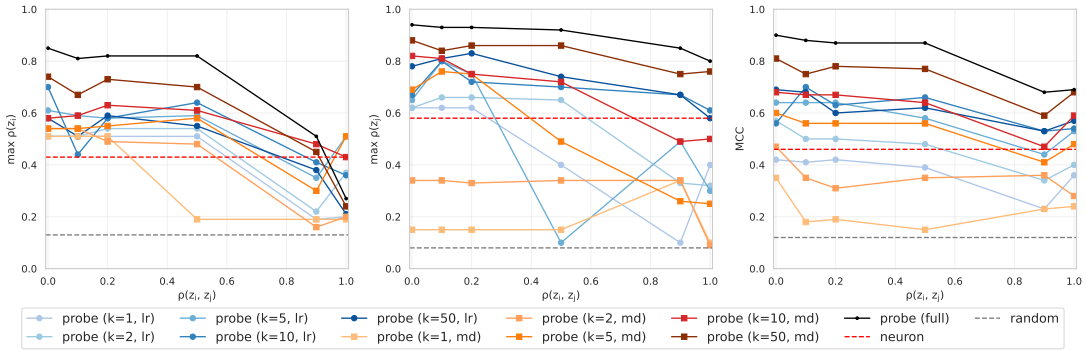

Figure 11: **MCC for the two most performant sparse probing methods from Gurnee et al. (2023) at various k.** Results for Pythia-70M shown here. The LR method achieves higher MCC at lower k, but MD overtakes LR at higher k.

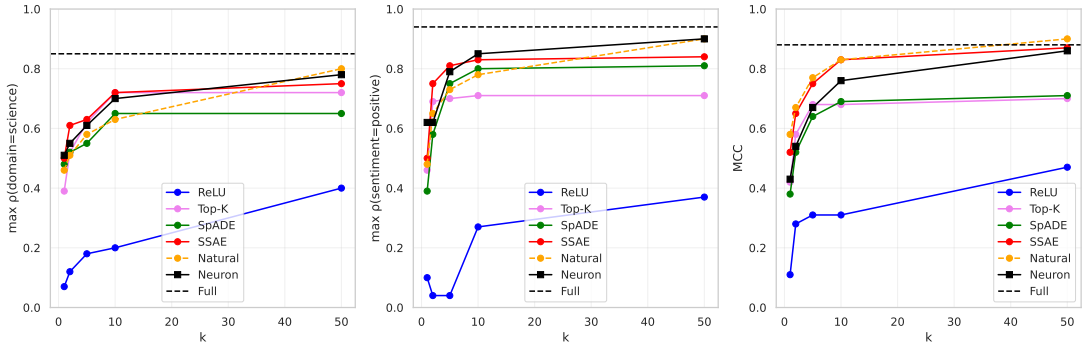

Figure 12: **Correlation coefficients between probe logits and concept labels for domain=science (left), sentiment=positive (middle), and MCC (right).** Results for Pythia-70M. We vary the number of dimensions k that the probe is allowed to have non-zero weights from. As with Gemma-2-2B, correlation coefficients tend to converge at around 10 dimensions. However, the neuron baseline is far more performant; perhaps this is because $k = 10$ represents a far greater proportion of the dimensions of $\mathbf{h}^\ell$ for Pythia than Gemma. Other trends are largely consistent with Figure 8.

All ideas originated from the authors. All quantitative results presented in the paper and all references were fully human-written. Any generated artifacts were verified and generally significantly changed by the authors when used.

We also used AI tools for assistance in coding; this primarily included debugging and planning assistance. Any generated artifacts were manually verified, and almost always significantly modified when used.

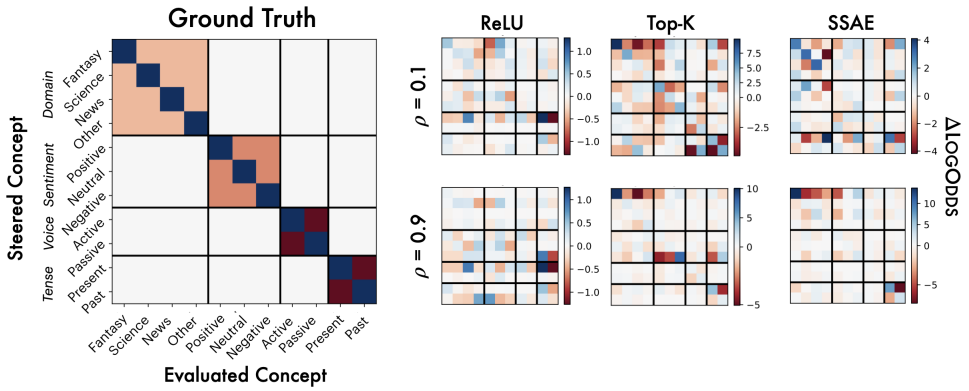

Figure 13: **The effect of steering a given concept (row) on the logit of another concept (column)**. Results for Gemma-2-2B. If concept representations are causally independent, we expect a heatmap that resembles the ground-truth: ΔLogOdds should be high on the diagonal, negative for within-concept pairs, and close to 0.0 for across-concept pairs. All SAEs demonstrate the expected diagonals, but also significant across-concept effects, indicating non-independence. Results are consistent across low and high correlations between concepts in the SAEs' training data.

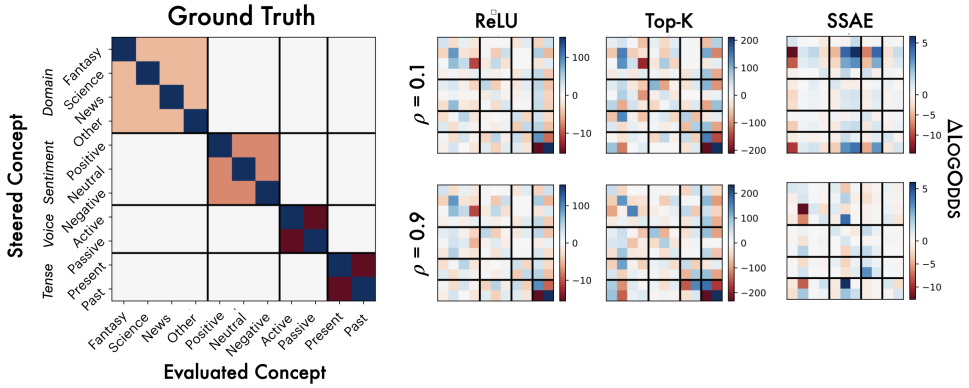

Figure 14: **The effect of steering a given concept (row) on the logit of another concept (column)**. Results for Pythia-70M when locating features using gradient attributions rather than activation correlations. If concept representations are causally independent, we expect a heatmap that resembles the ground-truth: ΔLogOdds should be high on the diagonal, negative for within-concept pairs, and close to 0.0 for across-concept pairs. All SAEs demonstrate the expected diagonals, but also significant across-concept effects, indicating non-independence. Results are consistent across low and high correlations between concepts in the SAEs' training data.

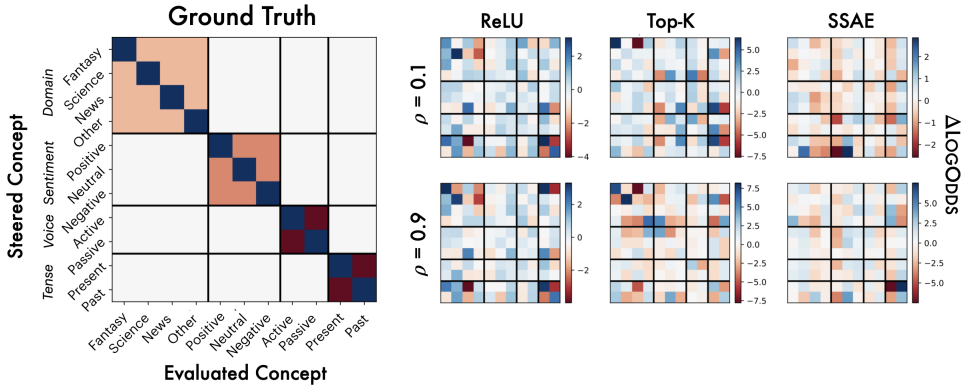

Figure 15: **The effect of steering a given concept (row) on the logit of another concept (column)**. Results for Gemma-2-2B when locating features using gradient attributions rather than activation correlations. If concept representations are causally independent, we expect a heatmap that resembles the ground-truth: ΔLogOdds should be high on the diagonal, negative for within-concept pairs, and close to 0.0 for across-concept pairs. All SAEs demonstrate the expected diagonals, but also significant across-concept effects, indicating non-independence. Results are consistent across low and high correlations between concepts in the SAEs' training data.

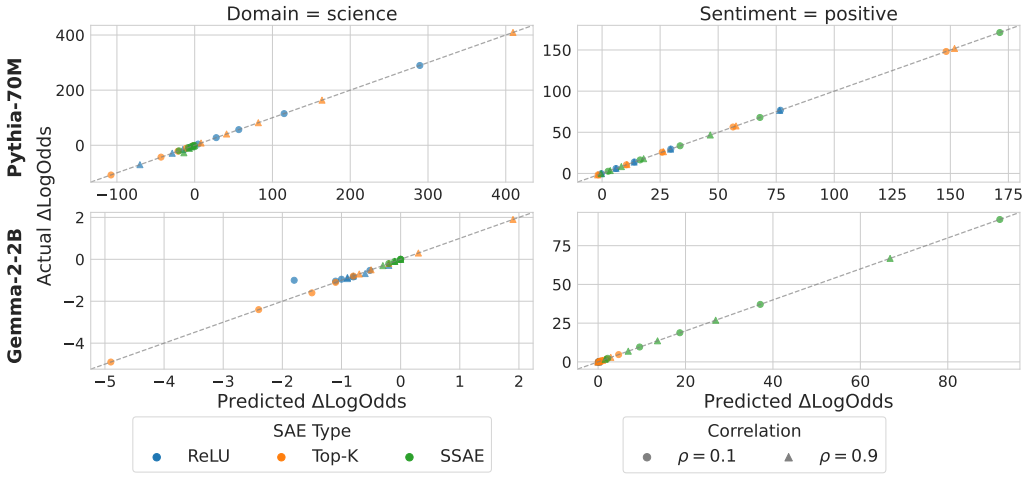

Figure 16: **Predicted ΔLogOdds(z_i) under disjointness assumptions vs. actual ΔLogOdds(z_i) when steering relevant feature $\hat{\mathbf{f}}_i$ and unrelated feature $\hat{\mathbf{f}}_j$.** Here, we select features using gradient attributions rather than activation correlations with the concept. $\hat{\mathbf{f}}_i$ and $\hat{\mathbf{f}}_j$ are typically disjoint, as indicated by the predicted change almost perfectly matching the actual change.

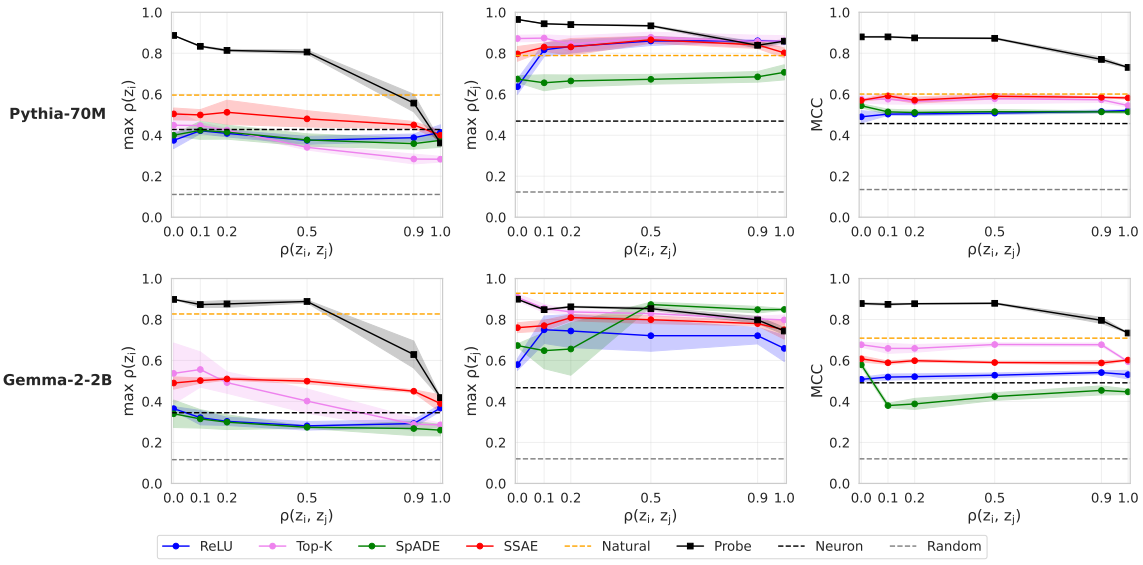

Figure 17: **Maximum correlation coefficient for tense=past (left), voice=passive (middle), and MCC (right) under varying correlational conditions.** Shaded regions represent 1 std. dev. across 3 training seeds. Ideal performance looks like a flat line at MCC=1. Probes (a supervised method) perform best, while Top-K SAEs and SSAEs are best among our SAEs. The Natural SAEs are especially effective at recovering tense=past.

Pythia-70M, ReLU		
It has been found that		
No steering	**Sentiment=positive ($\rho = 0$)**	**Sentiment=positive ($\rho = 1$)**
the first person to be the one who is the one who is the one	the most important part of the process of the process of the process of the	the presence of a high-fidelity material in the air-conditioning system is a very important factor

Gemma-2-2B, Top-K		
Once upon a time,		
No steering	**Domain=science ($\rho = 0$)**	**Domain=science ($\rho = 1$)**
a time when my children were very small, I bought a box of pencils (yes, that time). The box bore a very clear message: "It's never	there was a little girl who was born with a rare genetic disorder. She was born with a condition called "congenital heart disease." This condition is a birth defect that affects the heart's structure and function.	there was a brave little girl who was born with a heart condition. She was born with a hole in her heart,

Figure 18: **Qualitative examples of steering features in Pythia-70M and Gemma-2-2B.** We show results for two SAE architectures, ReLU and Top-K. We show the original output (No steering), the output after steering the top-attribution feature under no correlations between concepts during SAE training ($\rho = 0$), and the output after steering the top-attribution feature under a complete correlation between domain=science and sentiment=positive during SAE training ($\rho = 1$).