Interpreting CLIP with Sparse Linear Concept Embeddings (SpLiCE)

Usha Bhalla[*] Harvard University [a,b]
Alex Oesterling[*] Harvard University [b]
Suraj Srinivas Harvard University [b]
Flavio P. Calmon[†] Harvard University [b]
Himabindu Lakkaraju[†] Harvard University [b,c]

Abstract

CLIP embeddings have demonstrated remarkable performance across a wide range of multimodal applications. However, these high-dimensional, dense vector representations are not easily interpretable, limiting our understanding of the rich structure of CLIP and its use in downstream applications that require transparency. In this work, we show that the semantic structure of CLIP's latent space can be leveraged to provide interpretability, allowing for the decomposition of representations into semantic concepts. We formulate this problem as one of sparse recovery and propose a novel method, Sparse Linear Concept Embeddings (SpLiCE), for transforming CLIP representations into sparse linear combinations of human-interpretable concepts. Distinct from previous work, SpLiCE is task-agnostic and can be used, without training, to explain and even replace traditional dense CLIP representations, maintaining high downstream performance while significantly improving their interpretability. We also demonstrate significant use cases of SpLiCE representations including detecting spurious correlations and model editing. Code is provided at `https://github.com/AI4LIFE-GROUP/SpLiCE`.

1 Introduction

Natural images include complex semantic information, such as the objects they contain, the scenes they depict, the actions being performed, and the relationships between them. Machine learning models trained on visual data aim to encode this semantic information in their representations to perform a wide variety of downstream tasks, such as object classification, scene recognition, segmentation, or action prediction. However, it is often difficult to enforce explicit encoding of these semantics within model representations, and it is even harder to interpret these semantics post hoc to better understand what models may have learnt and how they leverage this information. Further, model representations can be brittle, encoding idiosyncratic patterns specific to individual datasets and modalities instead of general human-interpretable semantic information. Multimodal models have been proposed as a potential solution to this issue, and methods such as CLIP [1] have empirically been found to provide highly performant, semantically rich representations of image data. The richness of these representations is evident from their high performance on a variety of tasks, such as zero-shot classification and image retrieval [1], image captioning [2], and image generation [3]. However, despite their performance, it remains unclear how to quantify the semantic content contained in their dense representations. In this work, we answer the question: *can we decompose*

[*] Equal contribution, order by coin flip. [†] Equal contribution, alphabetic order.
[a] Kempner Institute for the Study of Natural & Artificial Intelligence
[b] School of Engineering and Applied Sciences [c] Harvard Business School

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

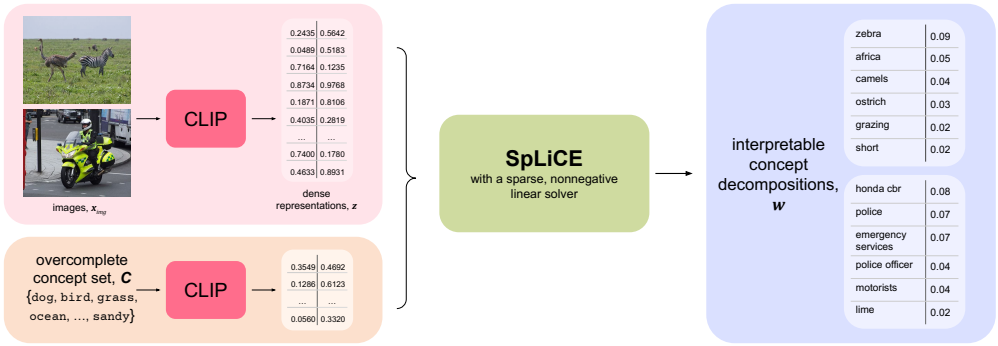

Figure 1: Visualization of SpLiCE, which converts dense, uninterpretable CLIP representations ($\mathbf{z}$) into sparse semantic decompositions ($\mathbf{w}$) by solving for a sparse nonnegative linear combination over an overcomplete concept set ($\mathbf{C}$).

CLIP embeddings into human-interpretable representations of the semantic concepts they encode? This can provide insight into the types of tasks CLIP can solve, the biases it may contain, and the manner through which downstream predictions are made.

Existing literature in areas such as concept bottleneck models [4], disentangled representation learning [5], and mechanistic intepretability [6] have proposed various approaches to understanding the semantics encoded by representations. However, these methods generally require predefined sets of concepts [7], data with concept labels [8], or rely on qualitative visualizations, which can be unreliable [9]. Similar to these lines of work, we aim to recover representations that reflect the underlying semantics of the inputs. However, distinct from these works, we propose to do this in a task-agnostic manner and without concept datasets, training, or qualitative analysis of visualizations.

Our method, SpLiCE, leverages the highly structured and multimodal nature of CLIP embeddings for interpretability, and decomposes CLIP representations via a semantic basis to yield a sparse, human-interpretable representation. Remarkably, these interpretable SpLiCE embeddings have favorable accuracy-interpretability tradeoffs when compared to black-box CLIP representations on metrics such as zero-shot accuracy. Our overall contributions are:

- In Sections 3 and 4, we formalize the sufficient conditions under sparse decomposition of CLIP is feasible, and introduce SpLiCE, a novel method that decomposes dense CLIP embeddings into sparse, human-interpretable concept embeddings.
- Our extensive experiments in Section 5 reveal that SpLiCE recovers highly sparse[1], interpretable representations with high performance on downstream tasks, while accurately capturing the semantics of the underlying inputs.
- In Section 6, we present two case studies for applying SpLiCE: spurious correlation detection, and model editing. Using SpLiCE, we uncover a spurious correlation in the CIFAR100 dataset, where we find the "woman" concept and the "swimwear" concept to be correlated owing to the prevalence of women in swimwear in CIFAR100.

2 Related Work

Linear Representation Hypothesis. In language modeling, the *linear representation hypothesis* suggests that many semantic concepts are approximately linear functions of model representations [10, 11, 12, 13, 14]. Recent work has also shown that multimodal models encode concepts additively, behaving like bags-of-words representations [15]. Relatedly, [16, 17] show that there exists a linear mapping between image and text embeddings in arbitrary models. Our work makes use of these distinct but related observations to convert dense CLIP representations to sparse semantic ones.

Concept Bottlenecks and Attribute Learning. Concept Bottleneck Models (CBMs) [18], and attribute-based models [19, 20, 21] learn intermediate representations of scores over concepts or image

[1] we recommend and use sparsity levels of $\sim$ 10-30 in practice

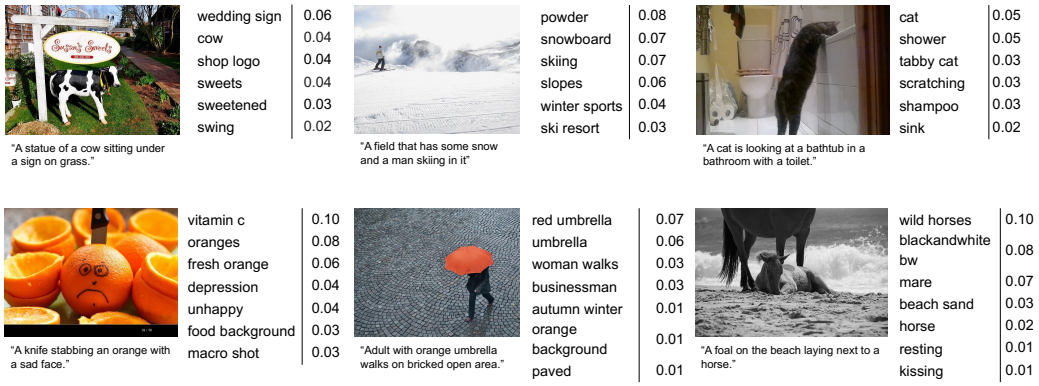

Figure 2: Example images from MSCOCO shown with their captions below and their concept decompositions on the right. We display the top seven concepts for visualization purposes, but images in the figure had decompositions with 7-20 concepts.

attributes for use with a final linear classification head, creating interpretable concept representations. However, these require expert-labeled concept or attribute datasets to train, which is expensive. Recent work on concept-bottlenecks for multimodal models avoids needing such labeled datasets, but still requires concept labels for specific tasks, which is obtained by querying large language models (LLMs) [22, 23, 24], making these methods task-specific and heavily reliant on the domain knowledge and subject to the biases of LLMs. On the other hand, SpLiCE uses a large-scale and overcomplete concept dictionary, avoiding dependence on training, specific domain knowledge, or a downstream task. Consequently, it can even be applied to understand unstructured, unsupervised image datasets in a label-free manner.

Mechanistic Interpretability and Disentanglement. Mechanistic interpretability explains representations through model activations, by labeling circuits and neurons in networks with feature visualization [6, 25] or by measuring concept activations and directions in latent space [26, 27, 7, 28, 29, 30]. Recent work [31] combines these methods, using dictionary learning to extract visual concept activations, whose semantics can be identified via feature visualization. Work in disentangled representation learning has developed architectures that capture independent factors of variation in data [8, 32, 33, 5, 34, 35], allowing for manual probing of disentangled representations for human-interpretable concepts. In both mechanistic interpretability and disentangled representation learning, methods typically rely on labeled concept sets, manual labeling of visualizations, or computationally intensive searches over data and latent representations or neurons to identify concepts. However, associating human-interpretable semantics with arbitrary neurons or latent directions is challenging, leading to the unreliability [9, 36] exhibited by such methods. Our approach side-steps this issue by decomposing CLIP representations into a predetermined set of concepts.

CLIP Interpretability. Many recent works leverage the semantic structure of CLIP and its text encoder to interpret representations. For example, [37], [38], and [39] construct concept similarity scores of image embeddings for use by downstream CBMs or probes, but these representations are not interpretable due to their lack of sparsity and the presence of negative concepts. Chen et al. [40] create a custom vision-language architecture with a sparse latent dictionary, but it requires training from scratch and cannot be used post-hoc to explain existing models. Gandelsman et al. [41] also leverage the text encoder of CLIP to explain components of the image embedding, but are limited to ViT architectures and take a mechanistic interpretability-style approach requiring a labeled text dataset. Chattopadhyay et al. [22] build concept bottlenecks for specific classification tasks by expressing CLIP image representations as a sparse linear combination of task-specific concept vectors. However, their decomposition includes negative concepts, reducing interpretability, and uses task-specific concept dictionaries. Grootendorst [42] generate textual topics of datasets through multimodal topic modeling, which cannot provide explanations of individual representations. Distinct from these works, SpLiCE is more interpretable due to its sparsity, overcompleteness, and non-negativity, and is task-agnostic, aiming to serve as a drop-in replacement for black-box CLIP representations without requiring training.

Table 1: Sanity checking the linearity of CLIP Embeddings.

	w_a	w_b	COSINE$(\hat{z}, z)$
IMAGENET	0.48 ± 0.09	0.45 ± 0.09	0.76 ± 0.05
CIFAR100	0.45 ± 0.08	0.42 ± 0.08	0.75 ± 0.03
MIT STATES	0.48 ± 0.09	0.45 ± 0.09	0.76 ± 0.05
COCO TEXT	0.59 ± 0.12	0.47 ± 0.12	0.88 ± 0.04

3 When do Sparse Decompositions Exist?

In this section, we aim to answer the question: *under what conditions can CLIP representations be decomposed into sparse semantic representations*? To do so, we must reason about both the properties of CLIP as well as the properties of the underlying data.

Notation. Let $\mathbf{x}^{\text{img}} \in \mathbb{R}^{d_i}$, $\mathbf{x}^{\text{txt}} \in \mathbb{R}^{d_t}$ be image and text data, respectively. Given the CLIP image encoder $f : \mathbb{R}^{d_i} \to \mathbb{R}^d$ and text encoder $g : \mathbb{R}^{d_t} \to \mathbb{R}^d$, we define CLIP representations in $\mathbb{R}^d$ as $\mathbf{z}^{\text{img}} = f(\mathbf{x}^{\text{img}})$ and $\mathbf{z}^{\text{txt}} = g(\mathbf{x}^{\text{txt}})$. Our method uses dictionary learning to approximate $\mathbf{z}^{\text{img}}$ with a concept decomposition $\mathbf{w}^* \in \mathbb{R}^c_+$ over a fixed concept vocabulary $\mathbf{C} \in \mathbb{R}^{d \times c}$. We define the resulting reconstruction of $\mathbf{z}^{\text{img}}$ from $\mathbf{C}$ and $\mathbf{w}^*$ as $\hat{\mathbf{z}}^{\text{img}}$.

The goal of our method is to approximate $f(\mathbf{x}^{\text{img}}) \approx \mathbf{C}\mathbf{w}^*$, such that $\mathbf{w}^*$ is non-negative and sparse, and in this section we formalize when this is possible. We begin by considering a data-generating process for coupled image and text samples. Specifically, we model the generative process parameterized by a k-dimensional latent concept vector $\omega \in \mathbb{R}^k_+$ and a random noise vector $\epsilon \in \mathbb{R}^l$ as

$$\mathbf{x}^{\text{img}} = h^{\text{img}}(\omega, \epsilon), \quad \mathbf{x}^{\text{txt}} = h^{\text{txt}}(\omega, \epsilon), \quad \omega \sim \rho, \quad \epsilon \sim \phi,$$

where ρ is a prior distribution over semantic concepts, ϕ is a prior distribution over nonsemantic concepts (such as camera orientation and lighting for images or arbitrary choices between synonyms for text), and $h^{\text{img}} : \mathbb{R}^{k+l} \to \mathbb{R}^{d_i}$, and $h^{\text{txt}} : \mathbb{R}^{k+l} \to \mathbb{R}^{d_t}$ represent the real-world data-generating process from latent variables (ω, ϵ) to images and text respectively. Here, each coordinate $\omega_i \in \mathbb{R}_+$ encodes the degree of prevalence of the i^{th} concept in the underlying data. We now list a set of sufficient conditions for our data-generating process and CLIP that admit a sparse decomposition of images into concepts.

Sufficient Conditions for Sparse Decomposition.

1. Images and text are sparse in concept space: for some $\alpha \ll k$, we have $\|\omega\|_0 \leq \alpha, \forall\, \omega \sim \rho$.
2. CLIP captures semantic concepts ω and not ϵ: $\forall \epsilon, \epsilon', f \circ h^{\text{img}}(\omega, \epsilon) = f \circ h^{\text{img}}(\omega, \epsilon')$ and similarly for h^{txt}.
3. CLIP is linear in concept space: $g \circ h^{\text{txt}}$ and $f \circ h^{\text{img}}$ are linear in ω.
4. CLIP image and text encoders are aligned: for a given ω, $f \circ h^{\text{img}}(\omega, \epsilon) = g \circ h^{\text{txt}}(\omega, \epsilon)$.

We emphasize that the goal of enumerating a set of sufficient conditions for sparse decomposition is not to claim that these exactly hold in practice, but rather to reason about when sparse decompositions– as done in this work–are appropriate. In the Appendix (Section A.1, Prop. 1) we formalize and prove this claim, but in the interest of simplicity we keep the discussion here informal. We note that many of these are natural; Assumption 1 reflects how real-world images and text are simple and rarely contain complex semantic content, and the CLIP training process optimizes for Assumption 2 and 4[2]. Of these, the most critical one is Assumption 3, which closely relates to the linear representation hypothesis [11], which we investigate below.

Sanity Checking CLIP's Linearity. We provide evidence for the third assumption, the linearity of CLIP, in a toy setting. We begin by asking the following question to confirm the general linearity of CLIP embeddings: "if two inputs are concatenated, does their joint embedding equal the average

[2]In practice we find that CLIP's image and text encoders are not fully aligned, so we apply a preprocessing step (Sec 4.1).

of their two individual embeddings?". For the image domain, we combine two images, x_a, x_b, to form their composition x_{ab} by placing x_a in the top left quarter and x_b in the bottom right quarter of a blank image. For the text domain, we simply append text x_b to text x_a to form x_{ab}. We then embed x_a, x_b, x_{ab} with CLIP to get z_a, z_b, z_{ab}. Solving the equation $w_a * z_a + w_b * z_b = z_{ab}$ for scalar weights w_a, w_b then allows us to assess the linearity of z_a, z_b, z_{ab}. We report w_a, w_b and the cosine similarity between $\hat{z}_{ab} = [z_a, z_b] \cdot [w_a, w_b]$ and z_{ab} in Table 1.

In general, we find that the composition of two inputs results in an embedding that is approximately equal to the average of the two input components, with w_a, w_b being very close to 0.5 across all datasets and for both modalities, providing preliminary evidence for the linearity of CLIP embeddings for both image and language.

4 Method

In this section, we introduce SpLiCE, a method for expressing CLIP's image representations as sparse, nonnegative, linear combinations of concept dictionary elements. We begin by framing this problem as one of sparse recovery. We then discuss our design choices, including how we choose the concept dictionary and how to address the modality gap between CLIP's images and text representations. Finally, we formalize the optimization problem used in this work.

4.1 Sparse Nonnegative Concept Decomposition

Our goal is to construct decompositions of dense CLIP representations that are human-interpretable, useful, and faithful. To do so, we formulate decomposition as a sparse recovery problem with three main desiderata. First, for the decompositions to be interpretable to humans they must be comprised of human interpretable atoms. We argue that language is a naturally interpretable interface for humans, and construct our concept vocabulary $\mathbf{C}$ out of 1- and 2-word atoms, such as "coffee", "silver", and "birthday party". Second, our decompositions must be simple and concise, which can be formulated as a sparsity constraint on the recovery. A large body of work in computational linguistics [43, 44, 45, 14], neuroscience [46, 47], and interpretability [48, 49, 30] have demonstrated that a human-aligned semantic model should be sparse in representation. Furthermore, [48] found that users can best understand explanations with fewer than 32 concepts while in linguistics, [50, 51, 52] find participants describe concepts and objects with up to 20 semantic properties, motivating our desiderata of sparsity. Third, our decompositions must be constructive, i.e., we must decompose representations in terms of their constituent concepts. For this reason, we require the weights of decompositions to be strictly nonnegative, to avoid having "negative" concept weights which do not always carry semantic meaning. Furthermore, prior work by Zhou et al. [30] has argued that *"negations of concepts are not as interpretable as positive concepts."* More specifically, while a small set of concepts have well-defined antonyms which may be viewed as their negative counterparts (``day'' $\leftrightarrow$ ``night''), negative concepts do not carry semantic meaning in general (``tiger'' $\leftrightarrow$??). Furthermore, we find that even when antonyms exist, they are not negatives of each other in CLIP latent space (see Appendix B.10). To avoid dependence on negative weights and ensure that all concepts are captured, we construct an overcomplete dictionary containing a wide range of concepts, including antonyms. We build on top of this literature and provide a semantic decomposition satisfying these properties suitable for multimodal models like CLIP.

Concept Vocabulary. Natural language is an intuitive, interpretable, and compact medium for communicating semantic information. Thus, we choose to represent the semantic content contained in CLIP embeddings as combinations of natural language semantic concepts, where we define concepts as semantic units that can be expressed concisely, by one- or two-word phrases. Given that CLIP is used in a wide variety of downstream applications and is trained without a specific task in mind, we want our concept dictionary to be task-agnostic and to span *all possible concepts CLIP could have learnt*. To construct this vocabulary, we consider the most frequent one- and two-word bigrams in the text captions of the LAION-400m dataset [53], the dataset that most CLIP variants are trained on. We filter the captions to remove any NSFW samples and prune our concept set such that no two concept embeddings have a cosine similarity greater than 0.9. We also remove bigrams highly similar (> 0.9 cosine similarity) to the average of their individual words. We finally choose the top $10,000$ most common single-word concepts and the top 5000 most common two-word concepts as our concept vocabulary. We note that this vocabulary offers distinct advantages over those used in prior works.

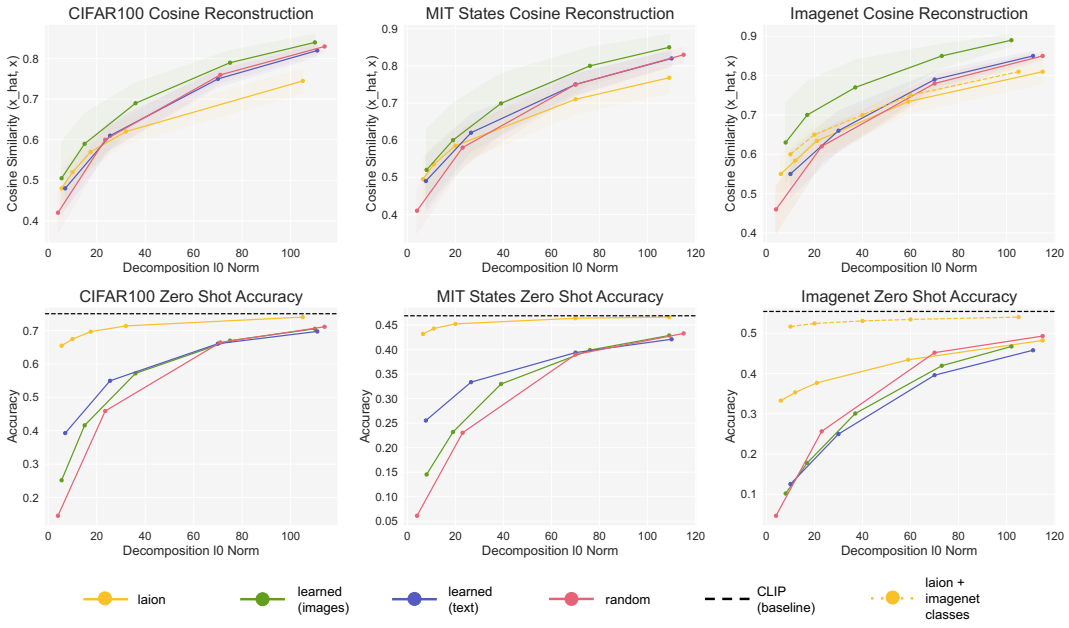

Figure 3: Performance of SpLiCE decomposition representations on zero-shot classification tasks (bottom row) and cosine similarity between CLIP embeddings and SpLiCE embeddings (top row). Our proposed semantic dictionary (yellow) closely approximates CLIP on zero-shot classification accuracy, but not on the cosine similarity. This indicates that SpLiCE captures the semantic information in CLIP, but not its non-semantic components, explaining both the high zero-shot accuracy and low cosine similarity. See §5.2 for discussion.

In particular, it is *task-agnostic*, meaning that the efficacy of the decomposition is (in principle) independent of individual datasets. Furthermore, this dataset imposes minimal priors from outside curators, such as human experts or LLMs [22, 23, 24]. This allows us to interpret data through the lens of CLIP, to understand the information encoded, including potential biases and mistakes.

Modality Alignment. In order to decompose images into nonnegative combinations of text concepts, we must ensure that our concept set spans the space of possible image embeddings. However, [54] show the existence of a modality gap in CLIP, where image and text embeddings can lie in non-identical spaces on the unit sphere. We empirically find that CLIP image and text embeddings exist on two cones, as the distribution of pairwise cosine similarities between pairs of MSCOCO images and pairs of MSCOCO text captions concentrate at positive values, whereas the distribution of pairwise cosine similarities across modalities concentrates closer to zero. (See Appendix Fig. 7). Not only does this prevent nonnegative decomposition, it also violates Assumption 4 from Section 3. To rectify this, we mean-center CLIP images with the image cone mean, estimated over MSCOCO ($\mu_{\mathbf{img}}$), and compute decompositions over the mean-centered concept vocabulary ($\mu_{\mathbf{con}}$). Note that the embeddings need to be re-normalized after centering to ensure they lie on the unit sphere. To convert our decompositions back into dense representations ($\hat{\mathbf{z}}^{\text{img}}$), we uncenter the normalized dense embeddings $\hat{\mathbf{z}}^{\text{img}}$ by adding the image mean back in and normalizing once again, to ensure they lie on the same cone as the original CLIP embeddings ($\mathbf{z}^{\text{img}}$).

Optimization Problem. Our optimization problem is formulated as follows. Let $\sigma(\mathbf{x}) = \mathbf{x}/\|\mathbf{x}\|_2$ be the normalization operation. Given a set of semantic concepts $\mathbf{x}^{\text{con}} = [\text{“}\texttt{dog}\text{”}, \text{“}\texttt{tabby cat}\text{”}, \text{“}\texttt{cloudy}\text{”}, \cdots]$, we construct a centered vocabulary $\mathbf{C} = [\sigma(g(\mathbf{x}_1^{\text{con}}) - \mu_{\text{con}}), \cdots, \sigma(g(\mathbf{x}_c^{\text{con}}) - \mu_{\text{con}})]$, where we recall that $g(\cdot)$ is the CLIP text encoder. Now, given the dictionary $\mathbf{C}$ and a centered CLIP embedding $\mathbf{z} = \sigma(\mathbf{z}^{\text{img}} - \mu_{\text{img}})$, we seek to find the sparsest solution that gives us a cosine similarity score of at least $1 - \epsilon$ for some small ϵ:

$$\min_{\mathbf{w} \in \mathbb{R}^c_+} \|\mathbf{w}\|_0 \text{ s.t. } \langle \mathbf{z}, \sigma(\mathbf{C}\mathbf{w}) \rangle \geq 1 - \epsilon. \tag{1}$$

As is standard practice, we relax the ℓ_0 constraint and reformulate this as a minimization of MSE with an ℓ_1 penalty, to construct the following convex relaxation[3] of Eq. (1):

$$\min_{\mathbf{w} \in \mathbb{R}^c_+} \|\mathbf{C}\mathbf{w} - \mathbf{z}\|_2^2 + 2\lambda\|\mathbf{w}\|_1. \tag{2}$$

Given the solution to the above problem $\mathbf{w}^*$, our reconstructed embedding is: $\hat{\mathbf{z}}^{\text{img}} = \sigma(\mathbf{C}\mathbf{w}^* + \mu_{\text{img}})$.

5 Experiments

In this section, we evaluate our method to ensure that SpLiCE decompositions are interpretable, performant, and accurately reflect the semantic content of representations.

5.1 Setup

Models. All experiments shown in the main paper are done with the OpenCLIP ViT-B/32 model [55] with results for an additional model in Appendix B.14. For all zero-shot classification tasks, we use the prompt template "A photo of a {}". **Datasets.** We use CIFAR100 [56], MIT States [57], CelebA [58], MSCOCO [59], and ImageNetVal [60] for our experiments with results for additional datasets in the Appendix (Section B.4)

Decomposition. For all experiments involving concept decomposition, we use sklearn's [61] Lasso solver with a non-negativity flag and an l_1 penalty that results in solutions with l_0 norms of 5-20 (around 0.2-0.3 for most datasets).We use a concept vocabulary chosen from a subset of LAION tokens as described in Section 4.1. Both image embeddings and dictionary concepts are centered and normalized as mentioned in Section 4.1, with the image mean used for centering computed over the MSCOCO train set and the concept mean computed over our chosen vocabulary.

5.2 Sparsity-Performance Tradeoffs

We assess the performance of SpLiCE decompositions by evaluating the reconstruction error in terms of cosine similarity between SpLiCE representations and CLIP embeddings, the zero-shot performance of SpLiCE decompositions, and the retrieval performance of SpLiCE embeddings. We compare the performance of decompositions generated from our semantic concept vocabulary to decompositions over random vocabulary and learned dictionary vocabulary baselines. All vocabularies are of size 15,000 concepts. The random vocabulary is sampled from a 512-dimensional normalized Gaussian distribution. The learned vocabularies are generated by using the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [62] to learn optimal dictionaries given our sparse recovery problem (optimizing Equation (1) for both $\mathbf{C}$ and w). Note that we learn separate dictionaries $\mathbf{C}_{\text{img}}$ and $\mathbf{C}_{\text{text}}$ to reconstruct MSCOCO image and text embeddings respectively. In Figure 3, we plot the cosine reconstruction and zero-shot accuracy of image decompositions with the various dictionaries. We evaluate probing performance (Tables 3, 4) and text-to-image and image-to-text retrieval in the Appendix (Figure B.3).

These results overall show SpLiCE efficiently navigates the interpretability-accuracy Pareto frontier and retains much of the performance of black-box CLIP representations with the semantic, human-interpretable LAION dictionary, significantly outperforming other dictionaries on semantic tasks such as zero-shot classification, probing, and retrieval. At the same time, we find that our semantic LAION dictionary does not result in accurate cosine similarity reconstructions of the original CLIP, often being on par with using random dictionaries. We believe this is because CLIP encodes both semantics of the underlying image and non-semantic "noise", which violates Assumption #2 in Section 3. Given that our SpLiCE decompositions only aim to encode semantics, they are unable to encode non-semantic aspects in the underlying representation, thus causing poor alignment in the cosine similarity sense, while simultaneously exhibiting excellent alignment on semantic tasks such as zero-shot accuracy. For ImageNet, we find that many classes are animal species that cannot easily be described by 1-2 words (e.g. '`red-breasted merganser`', '`American Staffordshire terrier`'). Adding these class labels to our concept dictionary increases performance significantly, as shown by the dotted yellow line in Figure 3.

[3]For more discussion on the relationship between Eq. (1) and Eq. (2), see Appendix, Sec. A.2

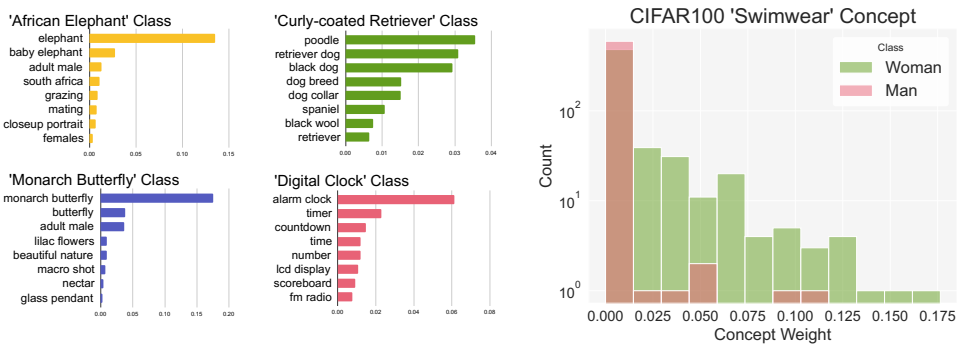

Figure 4: **Left:** SpLiCE decompositions of ImageNet '`African Elephant`', '`Curly-coated Retriever`', '`Monarch Butterfly`', '`Digital Clock`' classes. **Right:** Distribution of "`Swimwear`" concept in '`Woman`' and '`Man`' classes of CIFAR100.

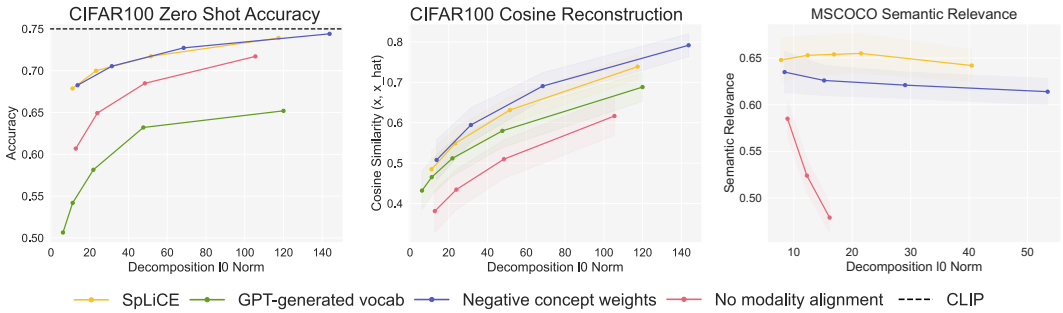

Figure 5: Ablation study evaluating the efficacy of SpLiCE design choices across three metrics: Zero-shot accuracy, cosine reconstruction, and semantic relevance of recovered tags. We find that all of our design choices, namely non-negativity, modality alignment, and usage of large task-agnostic dictionary are essential to performance. See §5.3 for discussion.

5.3 Ablation Studies

We perform ablation studies to evaluate the effectiveness of the design decisions of SpLiCE, including the choice of vocabulary, the nonnegativity of the decompositions, and the modality alignment by ablating each choice and observing the effect on three metrics: zero-shot accuracy on CIFAR100, cosine similarity between the reconstructed and original embeddings of CIFAR100, and semantic relevance on MSCOCO. The first two metrics are the same as those presented in Figure 3. We compute semantic relevance by tokenizing and filtering stop-words from the MSCOCO human-generated captions and embedding each token with CLIP. Then, we take all non-zero concepts output by SpLiCE and compute the Hausdorff distance between the sets of SpLiCE concepts and caption token embeddings. This essentially measures how aligned decompositions are with human captions. We observe that replacing our dictionary with the LLM-generated concept dictionary used by [22, 24, 23] significantly worsens the decomposition in terms of zero shot accuracy and cosine reconstruction. While allowing for negative concept weights improves cosine reconstruction marginally, it decreases the semantic relevance of the decompositions, as negative concepts frequently correspond to concepts not present in images, and as such, are unlikely to be represented by human captions. Finally, we see that modality alignment is necessary across all three metrics. Overall, these ablation studies show that each aspect of SpLiCE is necessary for creating human-interpretable, semantically relevant and highly performant decompositions.

5.4 Qualitative Assessment of Decompositions

Concept Decompositions for Images. We visualize SpLiCE decompositions to qualitatively assess the semantic content of the images they represent. In Figure 2 we provide six sample decompositions from MSCOCO with their corresponding captions. We display the top seven concepts for each

image and find that they generally well describe the semantics of the images. We also find that these qualitative examples yield interesting and unexpected insights into both CLIP and the data. In the top left image, we see that the decomposition includes the text present on the sign in the image, revealing that CLIP prioritizes text in images over objects. For the bottom left image, the decomposition correctly includes the concept "macro shot", revealing that CLIP encodes information regarding geometric perspective. The bottom right decomposition similarly features the concept "blackandwhite bw", indicating that CLIP encodes not only the objects present in images but also information about the lighting and color. Overall, these results suggest that SpLiCE may also be used as a zero-shot image tagging method to understand images.

Concept Histograms for Datasets. Beyond concept-based explanations of individual images, we propose that SpLiCE can be used to better understand and summarize collections of images, such as entire datasets. To compute concept decompositions of sets of images, we decompose each individual image and aggregate the results, which we use to generate concept histograms of the dataset. We visualize four concept histograms for the ImageNet classes 'African Elephant', 'Curly-coated Retriever', 'Monarch Butterfly', and 'Digital Clock', in Figure 4. These decompositions provide information about the distribution of the data and how CLIP represents it. For example, digital clocks are differentiated from analog clocks through the concepts "lcd display" and "countdown". Monarch butterflies are highly correlated with the concept "lilac flowers" in ImageNet, which we validated through manual inspection (nearly half of the monarch butterfly images in the validation set feature purple flowers). Interestingly, 'Curly-coated retrievers' are represented as combinations of "poodle", "retriever dog", and "black dog", which perfectly describe the main characteristics of them: black retrievers with poodle-textured fur.

6 Case Studies and Applications of SpLiCE

In this section, we present two example case studies using SpLiCE: (1) spurious correlation and bias detection in datasets and (2) debiasing classification models. We present additional case studies for (1) and (2), as well as (3) monitoring distribution shift in Appendix B.6, B.7, B.8 B.9. We also present results from a user study to evaluate the human interpretability of SpLiCE in Appendix B.1, where we find that users prefer explanations generated by SpLiCE over existing Concept Bottleneck Model-based methods.

Discovering Spurious Correlations in CIFAR100. Existing methods to detect spurious correlations in datasets generally require subgroup and attribute labels or rely on manual human inspection of images (see [63] for an overview), making it hard to scale to large datasets. SpLiCE, on the other hand, allows for fast automatic detection of such biases, without any labels, training, or even a task. To illustrate this, we study two classes of CIFAR100: 'man' and 'woman', in Figure 4. Upon decomposing these classes, we found that {"bra","swimwear" } were two of the top ten most common concepts in the 'woman' class. On the other hand, the only clothing-related concepts that appear in the top 50 most activated concepts for 'man' are {"uniform", "tuxedo", "apparel"}. We visualize a histogram of the concept weights on swimwear- and undergarment-related concepts {"swimwear", "bra", "trunks", "underwear"} across both the train and test sets, and find that these concepts are much more likely to be activated for women than men. Manual inspection of CIFAR100 verifies the trend highlighted by SpLiCE, where *at least 70 of the 600 images in the 'woman' class feature women in bikinis, underclothes, or even partially undressed*, revealing stereotype bias in this popular dataset. We provide a similar study of the concept "desert" with respect to the 'camel' and 'kangaroo' classes in CIFAR100 in Appendix B.6.

Model Editing on CelebA Attribute Classifiers. Concept-based representations unlock a key application: being able to intervene on and edit models. This edit can be performed in two equivalent ways: either on the concept representations themselves, where we can zero out a concept or on linear probes built upon the decompositions, where we can edit the weight matrix between concepts and class labels (similar to concept bottleneck models [18]). Here, we evaluate the efficacy of SpLiCE for these forms of model editing. Specifically, we consider two tasks on CelebA, classifying gender and whether the subject is wearing glasses. To test representation editing, we remove the concept of "eyewear" or "glasses" from CelebA image representations by zeroing out any weight placed on these concepts in our SpLiCE decompositions and evaluate classifier performance. We report the performance of zero-shot classification and linear probes over our SpLiCE representation in Table

Table 2: Evaluation of intervention on the concept 'Glasses' for the CelebA dataset. SpLiCE allows for surgical removal of information related to whether or not someone is wearing glasses, without impacting other features such as gender. (ZS = Zero Shot Accuracy)

	GENDER	GLASSES
ZS CLIP	0.98	0.91
ZS SPLICE	0.97	0.88
ZS INTERVENTION SPLICE	0.96	**0.69**
LINEAR PROBE	0.89	0.88
INTERVENTION PROBE	0.85	**0.59**

2. In both cases, we find that we can surgically target and remove information pertaining to glasses and reduce classifier performance while preserving information relevant to gender classification. We perform a similar experiment on the Waterbirds dataset [64] to remove spurious background signals in B.7.

7 Discussion

In this work, we show that the information contained in CLIP embeddings can be approximated by a sparse, linear combination of simple semantic concepts, allowing us to interpret representations via sparse recovery. We propose SpLiCE, a method to transform the dense, uninterpretable embeddings of CLIP into human-interpretable sparse concept decompositions.

We empirically demonstrate that SpLiCE allows for an adjustable tradeoff on the interpretability-accuracy Pareto frontier, enabling users to decide the loss in performance they are willing to incur for interpretability. Furthermore, we find that the improved interpretability of SpLiCE allows for users to diagnose and fix model mistakes, ideally increasing the effectiveness and performance of the overall system using a VLM. We then provide concrete use cases for SpLiCE: spurious correlation detection and model intervention and editing, showcasing the benefits of using interpretable embeddings with known semantic content. We highlight that SpLiCE embeddings can serve as post-hoc interpretations of CLIP embeddings and can even replace them to ensure full transparency.

Broader Impact. Similar to many works in the field of interpretability, our work provides greater understanding of the behavior of models, including but not limited to the broader implicit biases they perpetuate as well as mistakes made on individual samples. We believe this is particularly salient for CLIP, which is used in a variety of applications that are widely used in practice at this moment. We hope that insights gained from such interpretability allow users to make more informed decisions regarding how they interact with and use CLIP, regardless of their familiarity with machine learning or domain expertise in the task they are using CLIP for. We also highlight that SpLiCE can be used as a visualization-like tool for exploring and summarizing datasets at scale, allowing for easier auditing of spurious correlations and biases in both datasets and models.

Limitations. In this work, we use a large, overcomplete dictionary of one- and two-word concepts, however future work may wish to expand this dictionary or learn a dictionary over tokens (in discrete language space), to capture concepts with more than two words. This may also reduce the size of the dictionary and improve computation time. We note that this dictionary was constructed by looking at token frequency in the LAION-5B dataset, which has its own biases and may not correctly capture all the salient concepts that CLIP encodes. Despite this, we find that SpLiCE performs well on a variety of tasks while outperforming state-of-the-art concept dictionaries (Fig. 5, Appendix Fig. 13) and thus we believe LAION is a good dataset to generate a concept vocabulary from. We also note that this vocabulary can be easily modified by practitioners to consider additional concepts as needed for specific use cases. Finally, SpLiCE also uses an ℓ_1 penalty as the relaxation for ℓ_0 regularization, but future work may consider alternative relaxations or even binary concept weights.

Acknowledgements and Disclosure of Funding

This work is supported in part by the NSF awards IIS-2008461, IIS-2040989, IIS-2238714, FAI-2040880, and research awards from Google, JP Morgan, Amazon, Adobe, Harvard Data Science Initiative, and the Digital, Data, and Design (D^3) Institute at Harvard. AO is supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-2140743, and UB is funded by the Kempner Institute Graduate Research Fellowship. The views expressed here are those of the authors and do not reflect the official policy or position of the funding agencies.

References

[1] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021.

[2] Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021.

[3] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

[4] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR, 2017.

[5] Yoshua Bengio. Deep learning of representations: Looking forward. In *International conference on statistical language and speech processing*, pages 1–37. Springer, 2013.

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

[7] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018.

[8] Kyle Hsu, Will Dorrell, James CR Whittington, Jiajun Wu, and Chelsea Finn. Disentanglement via latent quantization. *arXiv preprint arXiv:2305.18378*, 2023.

[9] Robert Geirhos, Roland S Zimmermann, Blair Bilodeau, Wieland Brendel, and Been Kim. Don't trust your eyes: on the (un) reliability of feature visualizations. *arXiv preprint arXiv:2306.04719*, 2023.

[10] Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751, 2013.

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

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

[13] Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. A latent variable model approach to pmi-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399, 2016.

[14] Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah Smith. Sparse overcomplete word vector representations. *arXiv preprint arXiv:1506.02004*, 2015.

[15] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *The Eleventh International Conference on Learning Representations*, 2022.

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

[17] Ashish Seth, Mayur Hemani, and Chirag Agarwal. Dear: Debiasing vision-language models with additive residuals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6820–6829, 2023.

[18] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR, 2020.

[19] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE conference on computer vision and pattern recognition*, pages 951–958. IEEE, 2009.

[20] Lorenzo Torresani, Martin Szummer, and Andrew Fitzgibbon. Efficient object category recognition using classemes. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I 11*, pages 776–789. Springer, 2010.

[21] Neeraj Kumar, Alexander C Berg, Peter N Belhumeur, and Shree K Nayar. Attribute and simile classifiers for face verification. In *2009 IEEE 12th international conference on computer vision*, pages 365–372. IEEE, 2009.

[22] Aditya Chattopadhyay, Ryan Pilgrim, and Rene Vidal. Information maximization perspective of orthogonal matching pursuit with applications to explainable ai. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[23] Tuomas Oikarinen, Subhro Das, Lam M Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. *arXiv preprint arXiv:2304.06129*, 2023.

[24] Konstantinos Panagiotis Panousis, Dino Ienco, and Diego Marcos. Sparse linear concept discovery models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2767–2771, 2023.

[25] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001, 2020.

[26] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6541–6549, 2017.

[27] Ruth Fong and Andrea Vedaldi. Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8730–8738, 2018.

[28] Thomas McGrath, Andrei Kapishnikov, Nenad Tomašev, Adam Pearce, Martin Wattenberg, Demis Hassabis, Been Kim, Ulrich Paquet, and Vladimir Kramnik. Acquisition of chess knowledge in alphazero. *Proceedings of the National Academy of Sciences*, 119(47):e2206625119, 2022.

[29] Adriano Lucieri, Muhammad Naseer Bajwa, Stephan Alexander Braun, Muhammad Imran Malik, Andreas Dengel, and Sheraz Ahmed. On interpretability of deep learning based skin lesion classifiers using concept activation vectors. In *2020 international joint conference on neural networks (IJCNN)*, pages 1–10. IEEE, 2020.

[30] Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. Interpretable basis decomposition for visual explanation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134, 2018.

[31] Thomas Fel, Victor Boutin, Mazda Moayeri, Rémi Cadène, Louis Bethune, Mathieu Chalvidal, Thomas Serre, et al. A holistic approach to unifying automatic concept extraction and concept importance estimation. *arXiv preprint arXiv:2306.07304*, 2023.

[32] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, page 2, 2023.

[33] Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31, 2018.

[34] Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3): 287–314, 1994.

[35] Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000.

[36] Aleksandar Makelov, Georg Lange, and Neel Nanda. Is this the subspace you are looking for? an interpretability illusion for subspace activation patching. *arXiv preprint arXiv:2311.17030*, 2023.

[37] Mazda Moayeri, Keivan Rezaei, Maziar Sanjabi, and Soheil Feizi. Text-to-concept (and back) via cross-model alignment. In *International Conference on Machine Learning*, pages 25037–25060. PMLR, 2023.

[38] Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. *arXiv preprint arXiv:2205.15480*, 2022.

[39] Tian Yun, Usha Bhalla, Ellie Pavlick, and Chen Sun. Do vision-language pretrained models learn composable primitive concepts? *arXiv preprint arXiv:2203.17271*, 2022.

[40] Chen Chen, Bowen Zhang, Liangliang Cao, Jiguang Shen, Tom Gunter, Albin Madappally Jose, Alexander Toshev, Jonathon Shlens, Ruoming Pang, and Yinfei Yang. Stair: Learning sparse text and image representation in grounded tokens. *arXiv preprint arXiv:2301.13081*, 2023.

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

[42] Maarten Grootendorst. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*, 2022.

[43] Brian Murphy, Partha Talukdar, and Tom Mitchell. Learning effective and interpretable semantic models using non-negative sparse embedding. In *Proceedings of COLING 2012*, pages 1933–1950, 2012.

[44] Alona Fyshe, Partha P Talukdar, Brian Murphy, and Tom M Mitchell. Interpretable semantic vectors from a joint model of brain-and text-based meaning. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 2014, page 489. NIH Public Access, 2014.

[45] Alona Fyshe, Leila Wehbe, Partha Talukdar, Brian Murphy, and Tom Mitchell. A compositional and interpretable semantic space. In *Proceedings of the 2015 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 32–41, 2015.

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

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

[48] Vikram V Ramaswamy, Sunnie SY Kim, Ruth Fong, and Olga Russakovsky. Overlooked factors in concept-based explanations: Dataset choice, concept salience, and human capability. *arXiv preprint arXiv:2207.09615*, 2022.

[49] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[50] David P Vinson and Gabriella Vigliocco. Semantic feature production norms for a large set of objects and events. *Behavior Research Methods*, 40(1):183–190, 2008.

[51] Ken McRae, George S Cree, Mark S Seidenberg, and Chris McNorgan. Semantic feature production norms for a large set of living and nonliving things. *Behavior research methods*, 37 (4):547–559, 2005.

[52] Peter Garrard, Matthew A Lambon Ralph, John R Hodges, and Karalyn Patterson. Prototypicality, distinctiveness, and intercorrelation: Analyses of the semantic attributes of living and nonliving concepts. *Cognitive neuropsychology*, 18(2):125–174, 2001.

[53] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.

[54] Victor Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Y Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. *Advances in Neural Information Processing Systems*, 35:17612–17625, 2022.

[55] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip. July 2021. doi: 10.5281/zenodo.5143773. URL `https://doi.org/10.5281/zenodo.5143773`. If you use this software, please cite it as below.

[56] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[57] Phillip Isola, Joseph J. Lim, and Edward H. Adelson. Discovering states and transformations in image collections. In *CVPR*, 2015.

[58] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[59] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[60] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[61] 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. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

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

[63] Shirley Wu, Mert Yuksekgonul, Linjun Zhang, and James Zou. Discover and cure: Concept-aware mitigation of spurious correlation. In *International Conference on Machine Learning*, pages 37765–37786. PMLR, 2023.

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

[65] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[66] Li Fei-Fei, Robert Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.

[67] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010.

[68] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[69] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[70] Michael Stark, Jonathan Krause, Bojan Pepik, David Meger, James J Little, Bernt Schiele, and Daphne Koller. Fine-grained categorization for 3d scene understanding. *International Journal of Robotics Research*, 30(13):1543–1552, 2011.

Appendix

Summary of Appendix Results

- **A**. Further Details on the Method
  - **A.1**. When do Sparse Decompositions Exist?
  - **A.2**. Relationship between cosine similarity and MSE optimization
  - **A.3**. ADMM for batched on-device LASSO optimization
  - **A.4**. Effect of Modality Alignment
  - **A.5**. Experimental Details
- **B**. Additional Results
  - **B.1**. User Study for Human Interpretability
  - **B.2**. Performance of SpLiCE on Probing Tasks
  - **B.3**. Performance of SpLiCE on Retrieval Tasks
  - **B.4**. Additional Zero-Shot Results
  - **B.5**. Additional ImageNet Concept Histograms
  - **B.6**. Additional Case Study: Detecting Spurious Correlations
  - **B.7**. Additional Case Study: Spurious Correlation Intervention
  - **B.8**. Additional Case Study: Distribution Shift Monitoring
  - **B.9**. Additional Case Study: Distribution Shift Monitoring
  - **B.10**. Checking the Interpretability of Negative Concepts
  - **B.11**. Understanding the Image Mean for Modality Alignment
  - **B.12**. Choice of Concept Vocabulary
  - **B.13**. Concept Type Distribution
  - **B.14**. Experiments on Alternative CLIP Architecture

A Further Details on the Method

A.1 When do Sparse Decompositions Exist?

Proposition 1. *Given Assumptions 1-5, CLIP image embeddings f can be written as a sparse linear combination of text embeddings, i.e,*

$$f(\mathbf{x}^{img}) = \mathbf{C}^{txt}\mathbf{w}; \;\; s.t. \;\; \|\mathbf{w}\|_0 \leq \alpha$$

where $\mathbf{w} \in \mathbb{R}_+^k$, and $\mathbf{C}^{txt} \in \mathbb{R}^{d \times k}$, which is the text concept dictionary defined previously.

Proof. Any vector ω can be written as $\omega = \sum_{i=1}^k \omega_i \mathbf{e}_i$, where $\omega_i \in \mathbb{R}_+$, and $\mathbf{e}_i \in \mathbb{R}^k$ is a one-hot vector with one at the i^{th} co-ordinate. Thus we have

$$\begin{aligned} f(\mathbf{x}^{\text{img}}) =& f \circ h^{\text{img}}(\omega, \epsilon) = f \circ h^{\text{img}}(\omega) \quad \text{(Assumption 2)} \\ =& f \circ h^{\text{img}}\left(\sum_{i=1}^k \omega_i \mathbf{e}_i\right) = \sum_{i=1}^k \omega_i \underbrace{f \circ h^{\text{img}}(\mathbf{e}_i)}_{\mathbf{c}_i^{\text{img}}} \quad \text{(Assumption 3)} \end{aligned}$$

Here we define $\mathbf{c}_i^{\text{img}} = f \circ h^{\text{img}}(\mathbf{e}_i)$ as the 'image' concept basis vector; analogous to the text concept basis vector $\mathbf{c}_i^{\text{txt}} = g \circ h^{\text{txt}}(\mathbf{e}_i)$ already defined. Thus Assumption 2 implies the existence of a sparse decomposition of f in terms of 'image' concept vectors $\mathbf{c}_i^{\text{img}}$. Additionally, Assumption 1 ensures that this decomposition is sparse, as ω is sparse. So far, we have $f(\mathbf{x}^{\text{img}}) = \mathbf{C}^{\text{img}}\omega \;\; s.t. \;\; \|\omega\|_0 \leq \alpha$. From Assumption 4, the image concept vectors and text concept vectors are equal to each other, i.e, $\mathbf{c}_i^{\text{img}} = f \circ h^{\text{img}}(\mathbf{e}_i) = g \circ h^{\text{txt}}(\mathbf{e}_i) = \mathbf{c}_i^{\text{txt}}$. Finally, from Assumption 5, we have that the text concept vectors $\mathbf{c}_i^{\text{txt}}$ are given simply by word embeddings g of individual words.

Stringing these arguments together, we have that image representations $f(\mathbf{x}^{\text{img}})$ can be written as a sparse linear combination of vectors obtain from CLIP word embeddings $\mathbf{c}_i^{\text{txt}}$. We finally set $\mathbf{w} = \omega$, thus proving the assertion.

□

A.2 Relationship between cosine similarity and MSE optimization.

Recall our ℓ_1 relaxed cosine similarity optimization problem from Eqn. (1),

$$\min_{\mathbf{w}\in\mathbb{R}_+^c} \|\mathbf{w}\|_0 \text{ s.t. } \langle \mathbf{z}, \frac{\mathbf{Cw}}{\|\mathbf{Cw}\|_2} \rangle \geq 1 - \epsilon. \tag{3}$$

First we relax the ℓ_0 constraint to an ℓ_1 penalty.

$$\max_{\mathbf{w}\in\mathbb{R}_+^c} \langle \mathbf{z}, \frac{\mathbf{Cw}}{\|\mathbf{Cw}\|_2} \rangle - \lambda \|\mathbf{w}\|_1. \tag{4}$$

By observing that $||x-y||_2^2 = \langle x-y, x-y\rangle = \langle x, x\rangle + \langle y, y\rangle - 2\langle x, y\rangle$ and that $\mathbf{z}$, $\frac{\mathbf{Cw}}{\|\mathbf{Cw}\|_2}$ are unit-norm, maximizing the above inner product is equivalent to minimizing the euclidean norm,

$$\min_{\mathbf{w}\in\mathbb{R}_+^c} ||\frac{\mathbf{Cw}}{\|\mathbf{Cw}\|_2} - \mathbf{z}||_2^2 + 2\lambda||\mathbf{w}||_1. \tag{5}$$

This is a non-convex problem, but we can relax this problem to achieve better reconstruction in terms of euclidan distance as shown in Eqn. (2),

$$\min_{\mathbf{w}\in\mathbb{R}_+^c} \|\mathbf{Cw} - \mathbf{z}\|_2^2 + 2\lambda\|\mathbf{w}\|_1. \tag{6}$$

This problem will optimize euclidean distance between $\mathbf{Cw}$ and $\mathbf{z}$. Consider two vectors x, y on the unit sphere such that $\langle \frac{x}{||x||}, \frac{y}{||y||} \rangle > 0$. While any vector αy, $\alpha > 0$ will have the same cosine similarity score, the optimal vector in terms of euclidean distance to x is the vector αy such that $\alpha = \text{proj}_y(x)$, or in other words the projection of x onto y. Thus, solving for euclidean distance to approximate x will find αy which we must then normalize to find the unit-norm solution y. This explains the normalizing process described in Section 4.1.

Additionally, we can view Eqn. (6) as applying shrinkage to $\mathbf{Cw}$. Reconverting from euclidean norm to inner product, Eqn. (6) becomes

$$\max_{\mathbf{w}\in\mathbb{R}_+^c} \langle \mathbf{Cw}, \mathbf{z}\rangle - \frac{1}{2}\langle \mathbf{Cw}, \mathbf{Cw}\rangle - \lambda||\mathbf{w}||_1 = \langle \mathbf{Cw}, \mathbf{z}\rangle - \frac{1}{2}||\mathbf{Cw}||_2^2 - \lambda||\mathbf{w}||_1. \tag{7}$$

In conclusion, our optimization problem maximizes the inner product while imposing a shinkage penalty and sparsity penalty. Empirically, our reconstructions $\mathbf{Cw}$ are low-norm, so we normalize after solving to recover the unit-norm reconstruction.

A.3 ADMM for batched on-device LASSO optimization.

As each decomposition requires solving a LASSO optimization problem, we implement the Alternating Direction Method of Multipliers (ADMM) algorithm in Pytorch over batches with GPU support for efficient decomposition of large scale datasets over large numbers of concepts [65]. In practice, ADMM achieves primal and dual tolerances of $1\text{e}-4$ in fewer than 1000 iterations on a batch size of 1024. We present an empirical comparison beterrn LASSO and ADMM in 6, where we find both methods to be approximately equivalent.

Next we derive the iterates for our ADMM algorithm. Recall our optimization problem,

$$\min_{\mathbf{w}\in\mathbb{R}_+^c} \|\mathbf{Cw} - \mathbf{z}\|_2^2 + 2\lambda\|\mathbf{w}\|_1. \tag{8}$$

ADMM breaks down convex optimization problems into multiple sub-problems while penalizing the difference in solutions. We break Eqn. (8) into two subproblems, one solving the euclidean distance

objective and one solving the ℓ_1 and nonnegativity constraint. We let w denote the former solution, z the latter, and u tracks the difference between the two. Our ADMM iterates (w^k, z^k, u^k) are

$$w^{k+1} = \arg\min_w (f(w) + \frac{\rho}{2}||w^k - z^k + u^k||_2^2), \tag{9}$$

$$z^{k+1} = (S_{\lambda/\rho}(w^{k+1} + u^k))_+, \tag{10}$$

$$u^{k+1} = u^k + w^{k+1} - z^{k+1}, \tag{11}$$

where S_κ is a soft-thresholding function used to satisfy the LASSO constraints,

$$S_\kappa(a) := \begin{cases} a - \kappa, & a > \kappa \\ 0, & |a| \leq \kappa \\ a + \kappa, & a < -\kappa \end{cases} \tag{12}$$

As our optimization function $f(w)$ is quadratic, we can analytically compute w^{k+1} as

$$w^{k+1} = (2\mathbf{C}^T\mathbf{C} + \rho)^{-1}(\rho v + 2\mathbf{C}w), \tag{13}$$

where $v = z^k - u^k$. In our experiments we set $\rho = 5$, and stop when tolerances $\epsilon_{\text{prim}} = ||x^{k+1} - z^{k+1}||_2$, $\epsilon_{\text{dual}} = ||\rho(z^{k+1} - z^k)||_2$ are less than $1\text{e} - 4$. Over a batch, we iterate until every solver in the batch has reached the above tolerances.

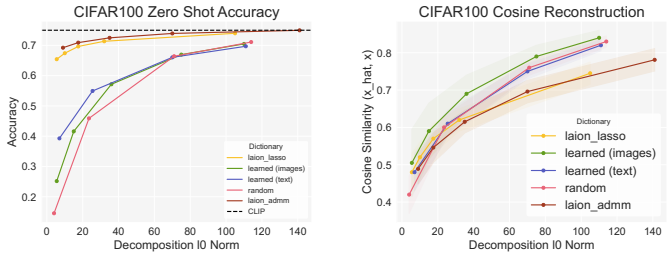

Figure 6: Comparison of ADMM (maroon) and LASSO (yellow) for solving the SpLiCE objective on zero shot accuracy (left) and cosine reconstruction (right) on CIFAR100. Both methods are approximately equal.

A.4 Effect of Modality Alignment

We take MSCOCO images and captions, embed them with CLIP, and compare the cosine similarity between modalities and inter-modality. Before mean-centering and renormalizing, the similarity within modalities is high, with an average of around 0.3. This indicates that the image and text embeddings do not span the entire unit-sphere but rather lie on two cones. However, the similarity across modalities has an average concentrating around zero, indicating that these two cones are non-overlapping. However, after mean-centering and normalizing, we observe that the average cosine similarity for images, text, and between images and text becomes zero and the modalities are aligned.

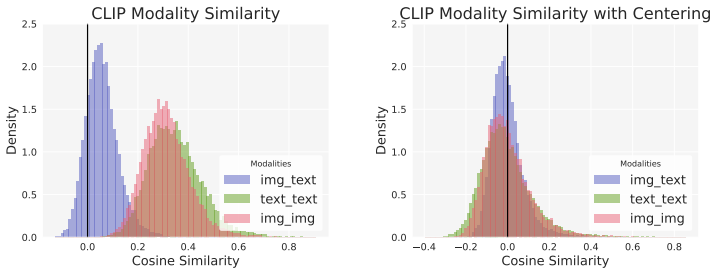

Figure 7: Average cosine similarity across pairs of image-text, image-image, and text-text data from MSCOCO. After aligning modalities, the distribution of similarities is centered around zero.

A.5 Experimental Details

All experiments are able to be performed on a single A100 GPU to run fast inference with CLIP. After embedding the concept dictionary, all computation can be performed on a CPU. Code is made available at `https://github.com/AI4LIFE-GROUP/SpLiCE`.

B Additional Results

B.1 User Study for Human Interpretability

We present results from a user study in 8 to assess the human interpretability of SpLiCE. We base our study off of that performed by [23] to evaluate Label-Free Concept Bottleneck Models (LF-CBMs). We benchmark our method against LF-CBMs and IP-OMP [22]. We provided users with twenty randomly chosen, correctly predicted images from ImageNet and explanations from two different methods comprising the top six most important concepts for every image. We then asked users to evaluate and compare the different concept-based explanations for (1) their relevance to the provided image inputs, (2) their relevance to model predictions, and (3) their informativeness on Likert scales from 1 to 5. We found that users significantly preferred explanations generated by SpLiCE to the two baselines for relevance to the images and informativeness, with significance determined via a one-sample two-sided t-test and a threshold of p=0.01. We also highlight that our method is able to produce similar/better concept decompositions, in terms of human interpretability, than the baselines without needing to train a classification probe or use class labels for concept mining, both of which are computationally expensive. This user study was ruled exempt by our institution's IRB, as no risks were posed to the users. Participants were able to opt out at any time, and no questions were asked regarding the participants themselves.

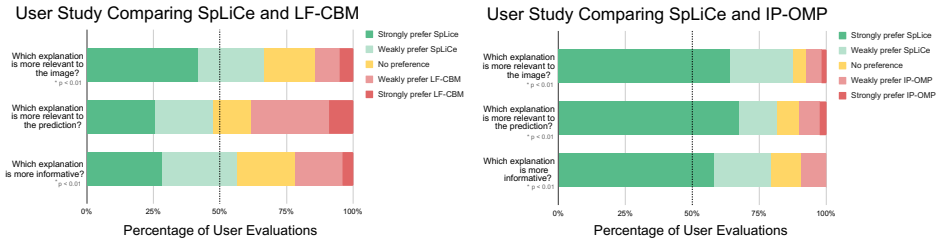

Figure 8: Results of a user study evaluating SpLiCE, LF-CBM, and IP-OMP in the style of the user study from LF-CBM. Overall, we find that explanations generated by SpLiCE are deemed more relevant to the image, relevant to the prediction, and more informative than prior methods.

B.2 Performance of SpLiCE on Probing Tasks

We evaluate the performance of the decompositions on probes trained on both regular CLIP embeddings as well as decomposed CLIP embeddings for CIFAR100 in 3 and MIT States in 4. We consider two scenarios: a probe trained on CLIP embeddings and tested on SpLiCE embeddings of various sparsities (shown in row `CLIP Probe`), and a probe both trained and evaluated on SpLiCE embeddings (shown in row `SpLiCE Probe`). We report mean over three runs, with standard deviations for each experiment being less than 0.005. We find that SpLiCE representations closely match the performance of dense CLIP embeddings, with a slight drop in performance when probes are trained directly on SpLiCE embeddings rather than trained on CLIP embeddings and evaluated on SpLiCE embeddings for CIFAR100.

Table 3: Evaluation of Probing Performance on CIFAR100

	$l_0 = 3$	$l_0 = 6$	$l_0 = 23$	$l_0 = 117$	CLIP
SpLiCE Probe	0.95	0.95	0.95	0.95	–
CLIP Probe	0.96	0.96	0.97	0.97	0.97

Table 4: Evaluation of Probing Performance on MIT States

	$l_0 = 4$	$l_0 = 7$	$l_0 = 27$	CLIP
SpLiCE Probe	0.883	0.883	0.882	–
CLIP Probe	0.883	0.883	0.884	0.883

B.3 SpLiCE Performance on Retrieval Tasks

We test the performance of SpLiCE embeddings on text-to-image and image-to-text retrieval tasks. We evaluate retrieval over various 1024 sample subsets of MSCOCO, and assess recall performance for the top-k closest embeddings of the opposite modality for $k = \{1, 5, 10\}$. We find that our semantic concept dictionaries outperform all baselines when decomposition sparsity is high, but that dictionaries learned over images perform slightly better for text to image retrieval when decompositions have greater than 30 nonzero concepts.

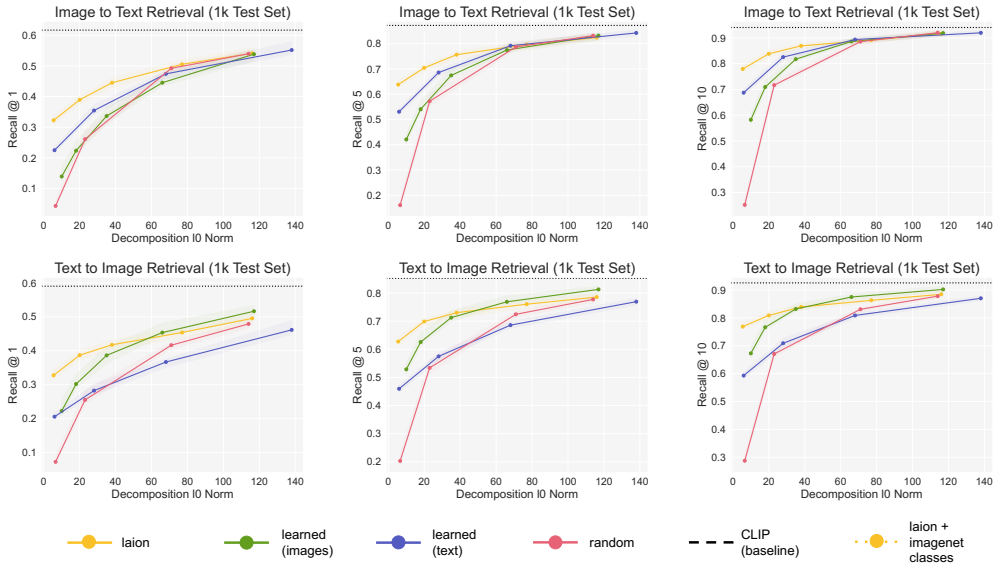

Figure 9: Top-1 , 5, 10 performance of SpLiCE representations on image-to-text (top) and text-to-image (bottom) retrieval on MSCOCO.

B.4 Additional Zero-Shot Results

We present additional results comparing SpLiCE reconstructed vectors and CLIP embeddings on the Caltech101 [66], SUN397 [67], STL10 [68], and VOC2007 [69] datasets in 5. We use SpLiCE decompositions with sparsities of 20-35, and we find that they are comparable to the unaltered CLIP embeddings.

Table 5: Additional zero-shot accuracy on baselines from the CLIP paper, for decompositions of sparsity 20-35. Note that at human-interpretable levels of sparsity, we see a minor drop in performance.

	Caltech101	SUN397	STL10	VOC 2007
CLIP Reported	0.88	0.63	0.97	0.83
CLIP Implemented	0.90	0.67	0.96	0.92
SpLiCE	0.86	0.66	0.96	0.83

We further explore the performance of SpLiCE decompositions in the limit as they approach the sparsity of the baseline CLIP embeddings (512). We find that SpLiCE completely recovers CLIP zero-shot accuracy at this limit, as shown in 6.

Table 6: Zero shot performance at sparsity 512. Note that SpLiCE completely recovers baseline CLIP zero shot accuracy.

	CIFAR100	MITSTATES	IMAGENET
CLIP BASELINE	0.750	0.469	0.552
SPLICE (512)	0.768	0.474	0.552

B.5 Additional ImageNet Concept Histograms

We present concept histograms for the top seven concepts of five more ImageNet classes: {'Face Powder', 'Feather Boa', 'Jack-O'-Lantern', 'Kimono', 'Dalmation'}, similar to Figure 10. These decompositions give insights both into the distribution of each class as well as some biases of CLIP. For example, for the class 'Face Powder', the concept "benefit" is the fifth most common concept, and it is indeed a common cosmetic brand name in the images. For the 'Dalmation' class, we see that the decompositions consists of concepts relating to dogs and black and white spots, which together make up the high-level concept of a dalmation. Finally, for the class 'Kimono', the concept "doll" is the seventh most common, although all of the images in the 'Kimono' class were of real humans, not of dolls. This highlights an implicit bias in CLIP's representations or in the descriptions of people wearing kimonos in CLIP's training set.

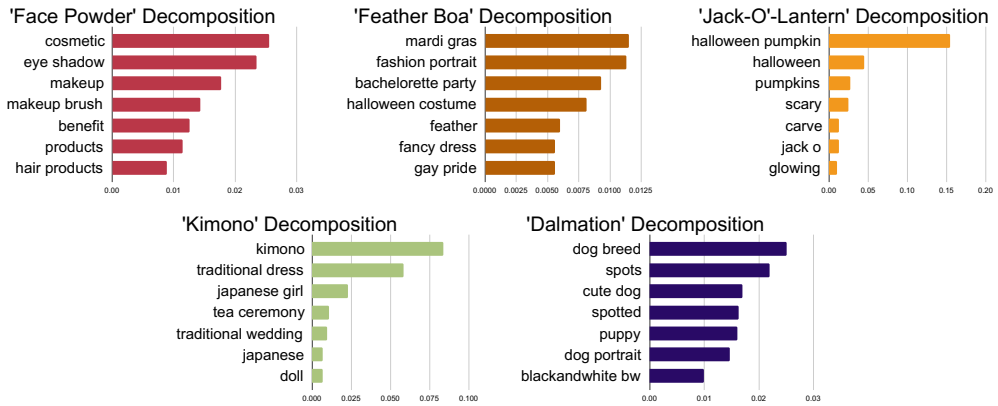

Figure 10: Example concept histograms of various ImageNet classes. The top seven concepts for each class are visualized along with their relative weighting, with the average ℓ_0 norm of individual sample decompositions also being 7.

B.6 Additional Case Study: Detecting Spurious Correlations

We present an additional case study for detecting spurious correlations in CIFAR100. In particular, we look at the prevalence of the spurious concept "desert" in the classes 'camel' and 'kangaroo' in Figure 11. We observe that camels are more frequently pictured in the desert, creating a spurious signal that may be leveraged by downstream classifiers. This figure provides an additional example of how we can understand biases and trends in data with SpLiCE decompositions.

B.7 Additional Case Study: Spurious Correlation Intervention

We further test the ability of SpLiCE to enable intervention on intermediate representations and linear classifiers by attempting to remove information pertaining to spurious signals. In particular, we consider the Waterbirds dataset [64], which spuriously correlates landbirds with land backgrounds, resulting in trained classifiers performing poorly on waterbirds on land. We thus remove information about whether or not birds are on land backgrounds by ablating concept weights on "bamboo",

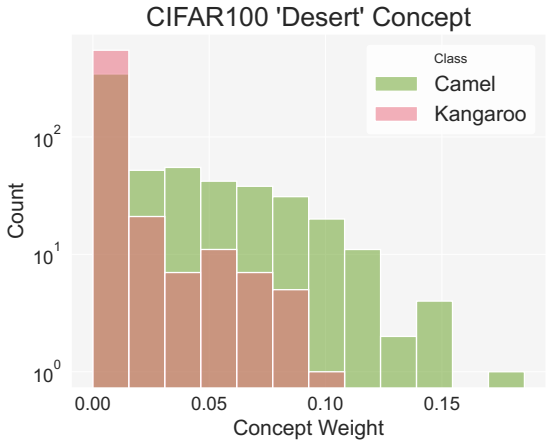

Figure 11: Distribution of “Desert" concept in ‘Camel’ and ‘Kangaroo’ classes of CIFAR100.

Table 7: Evaluation of intervention on spurious correlations for Waterbirds dataset. Removing information about land backgrounds improves worst-case subgroup performance.

	LANDBIRDS ON LAND	WATERBIRDS ON LAND
LINEAR PROBE	0.98	0.48
INTERVENTION PROBE	0.97	**0.60**

“forest”, “hiking”, and “rainforest” as well as any bigrams containing the word “forest,” as shown in Table 7. This significantly improves worst-case subgroup performance for waterbirds on land from 0.48 to 0.60.

For both this experiment and the intervention on CelebA described in the main paper, we train linear probes using the LogisticRegressionClassifier module in scikit-learn using an ℓ_1 penalty.

B.8 Additional Case Study: Distribution Shift Monitoring

We present a final case study using SpLiCE to monitor distribution shift. This can help identify differences between training and inference distributions or evaluate how a continually sampled dataset changes over time. In this experiment we consider the Stanford Cars dataset [70], which contains photos of cars from 1991 to 2012, including their make and year labels. By decomposing photos of cars from each year, we can view how the distribution changed yearly. We visualize the weights of the concepts “convertible" and “yellow" from our decompositions, as well as the actual percentage of cars from each year that were convertibles or yellow in Figure 12. Note the right-hand y-axis, corresponding to the weight of the given concept c_i over the sum of the weights of all concepts $\sum_i c_i$, does not have a meaningful unit of measure or scale. We find that the trends in the groundtruth concept prevalence generally closely match that of the predicted/decomposed concepts, allowing us to visualize which years convertibles or yellow cars were popular or out-of-distribution with respect to other years. Most notably, we see that SpLiCE picks up on the out-of-distribution rise in popularity of brightly colored sports cars in the early 2000s.

B.9 Additional Case Study: Distribution Shift Monitoring

To further verify that SpLiCE allows for identification and tracking of distribution shift, we study the Waterbirds dataset, which is known to have differently balanced train, valodation, and test splits. To identify distribution shifts, we can simply look at the norm of the difference between the class decompositions of the two classes for each splot, as shown in 8. We find that the validation and test splits are much more similar than the training and validation splits or the training and test splits, which can be verified by the construction process of the Waterbirds dataset.

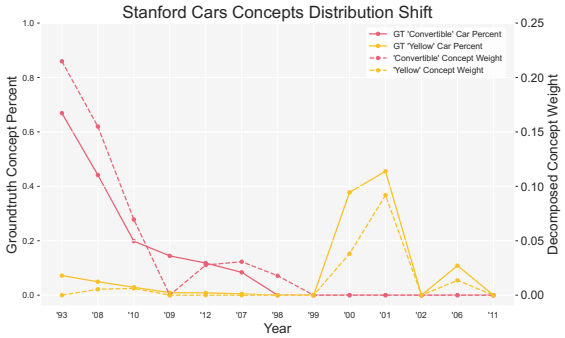

Figure 12: Visualization of the presence of convertibles (pink lines) and yellow cars (yellow lines) in Stanford Cars over time. SpLiCE concept weights (dotted) closely track the groundtruth concept prevalence (solid) for both concepts.

Table 8: Study of the differences in distributions between train, validation, and test splits of Waterbirds. The validation and test splits are much more similar to each other than they are to the train split.

	TRAIN, VAL	TRAIN, TEST	VAL, TEST
CLASS LANDBIRD	0.0182	0.0182	**0.005**
CLASS WATERBIRD	0.0229	.0188	**0.009**

We also find that the most weighted concept in the 'landbird' class of the train split is "bamboo" but the corresponding weight for "bamboo" in the 'waterbird' class is much lower. The "bamboo" concept weight for both classes and all splits is shown below, where we see that the validation and test splits are very similar and mostly evenly balanced, whereas the train split is highly unbalanced.

B.10 Checking the Interpretability of Negative Concepts

We take a set of 71 concept-antonym pairs from the MIT States dataset and embed the terms in CLIP. With and without concept centering, we observe that these concept-antonym pairs have an average cosine similarity well above -1, indicating that CLIP does not place antonyms in opposite directions, as shown in 10. Next, we take our concept dictionary and prepend "not" to all of the words and compare the average cosine similarity between concept and not-concept pairs. Similarly, we observe that with and without centering, concept and not-concept pairs are highly similar. Note that the average similarity for true pairs of images and text in MSCOCO is less than the similarity between concepts and not-concepts with and without centering.

B.11 Understanding the Image Mean for Modality Alignment

In order to empirically check that the mean centering of images does not result in a loss of information, we decompose the img mean, μ_{img}, that we used for all experiments. If we decompose it with uncentered concepts, the following concepts are highlighted: {"closeup", "flickr", "posed"}. The decomposition with centered concepts results in the following concepts: {"flickr", "posed", "pics", "angle view", "last post"}. These concepts all seem to be generally related to images, with minimal other semantic information, suggesting that centering does not remove any discriminative semantic content of embeddings, but simply removes information about the modality.

Table 9: Study of the prevalence of the concept "bamboo" in the different classes and splits of Waterbirds.

	TRAIN	VAL	TEST
CLASS LANDBIRD	**0.0196**	0.010	0.010
CLASS WATERBIRD	**0.0007**	0.008	0.008

Table 10: Evaluation of the similarity of antonyms and negative concepts in CLIP.

	PAIRWISE COSINE SIMILARITY (WITHOUT CONCEPT CENTERING)	PAIRWISE COSINE SIMILARITY (WITH CONCEPT CENTERING)
CONCEPT AND ANTONYM	0.7176 ± 0.1109	0.1366 ± 0.2197
CONCEPT AND "NOT" CONCEPT	0.8661 ± 0.0498	0.6130 ± 0.0498

B.12 Choice of Concept Vocabulary

We perform a simple ablation study to assess the sensitivity of our method to choices in concept vocabulary. We collect a second vocabulary in the same exact manner as the LAION vocabulary from the MSCOCO caption dataset. We consider both the top 10k and top 5k most common words for both, and repeat the zero-shot accuracy and reconstruction cosine similarity experiments from Section 5.2 on CIFAR100. We see that the MSCOCO10k and LAION10k vocabularies perform almost exactly the same for both metrics. The smaller vocabularies perform the same for cosine reconstruction but underperform the 10k vocabularies for zero-shot classification tasks.

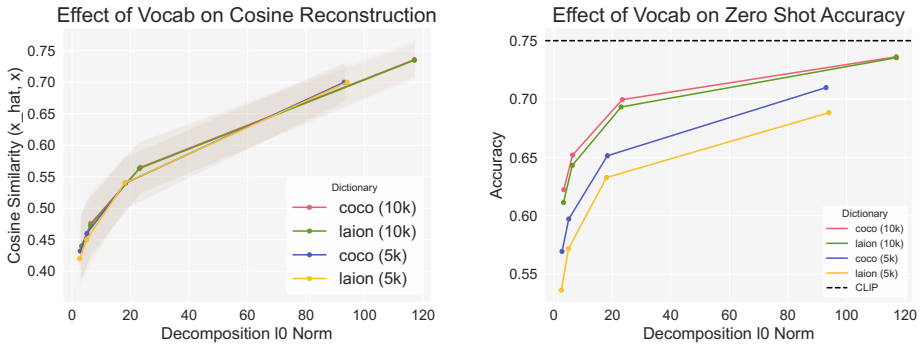

Figure 13: Change in SpLiCE performance when considering another semantic concept dictionary derived from MSCOCO as well as a smaller concept vocabulary.

B.13 Concept Type Distribution

In order to better understand any biases produced by the decomposition process or that CLIP itself has, we visualize the types of concepts most commonly activated across multiple datasets, labelling them by part of speech in Figure 14. We see that nouns are by far the most common concepts across datasets, indicating that both CLIP and the decompositions are highly object centric. Note that the low weight on verbs and adjective is due to far fewer concepts of those types being activated (low l_0 norm) as well as the weight upon those concepts being significantly smaller (low l_1 norm). We hypothesize that the information in many adjective and verbs can actually be encoded into the noun itself, resulting in this phenomenon. For example, the concept "lemon" is a more succinct form of "yellow" and "fruit".

B.14 Experiments on Alternative CLIP Architecture

We present cosine reconstruction and zero-shot accuracy experiments with an alternative CLIP architecture from OpenAI with a ResNet50 backbone for the vision encoder. Note that these experiments were done with a 10000 size vocabulary of only one-word concepts. We find that results are similar to those presented in 3, save for OpenAI's ResNet50 CLIP performing much worse than OpenCLIP's ViT B/32 backbone in general.

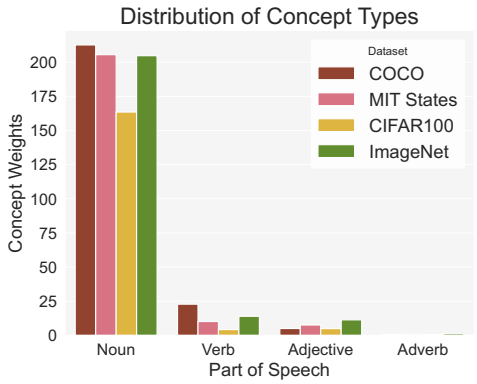

Figure 14: SpLiCE decompositions are mostly comprised of nouns across multiple datasets.

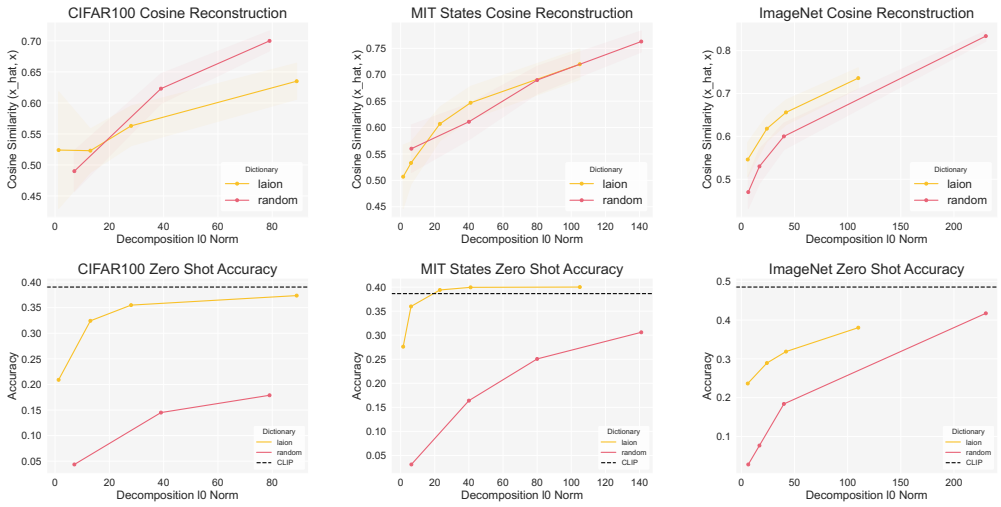

Figure 15: Performance of SpLiCE decomposition representations on zero-shot classification tasks (bottom row) and cosine similarity between CLIP embeddings and SpLiCE embeddings (top row) for OpenAI's ResNet50 CLIP model.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Yes, we introduce a novel method and provide comprehensive experiments demonstrating its utility on downstream tasks as well as various case studies in Sections 5 and 6.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: In the appendix we note the limitations of our work **??**.

   Guidelines:

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

3. **Theory Assumptions and Proofs**

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

Answer: [Yes]

Justification: We provide a detailed set of assumptions, proposition and proof sketch in the main paper section 3 and a full proof in section A.1

Guidelines:

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

4. **Experimental Result Reproducibility**

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

Answer: [Yes]

Justification: We disclose all information required to reproduce the main results by providing details on the exact methods and metrics used and providing code to implement SpLiCE . In our experiments (Sec 5) we disclose our datasets. Then, to replicate results, we describe how to compute zero-shot accuracy, cosine similarities, as well as how to generate concept histograms over classes and concepts as shown in (Figure 4.) These are also included in our code. We also include additional experimental details in the Appendix.

Guidelines:

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

In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: We use publicly-available datasets, MSCOCO, CIFAR100, MIT States, Imagenet, CelebA, Waterbirds, and Stanford Cars and include code to replicate our results (link at the end of abstract).

   Guidelines:

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

6. **Experimental Setting/Details**

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

   Answer: [Yes]

   Justification: We specify hyperparameters and other implementation details in 5.1. We describe how we optimize for Eqn. (2) using scikit-learn, as well as include an implementation in our codebase (linked in the abstract).

   Guidelines:

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

7. **Experiment Statistical Significance**

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

   Answer: [Yes]

   Justification: Throughout the paper we report error bars when possible. For instance, when computing reconstruction error we are able to report error bars (Fig. 3). However, for zero-shot error, this metric is fixed for any dataset so we do not report error bars there. When possible we run experiments 5 times and include error bars.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

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

   Answer: [Yes]

   Justification: We include details on compute resources in the Appendix A.5

   Guidelines:

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

9. **Code Of Ethics**

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

   Answer: [Yes]

   Justification: We do not have human subjects nor do we collect additional data. When conducting semantic decomposition, there is a risk that we include harmful semantics with our large dictionary. However, as evidenced in Figure 4, our method actually uncovers bias to help reduce it in the future rather than perpetuating bias. Semantic decompositions allow for a more critical view of our machine learning systems and datasets. In addition, we have open-sourced SpLiCE with essential elements for reproducibility to ensure public governance of the system.

   Guidelines:

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

10. **Broader Impacts**

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

    Answer: [Yes]

    Justification: We include a discussion of broader impact in the (Appendix **??**)

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: As our method does not release data, nor does it include a generative model, we do not have a high risk for misuse. Thus, we do not have any special safeguards on our method.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We properly cite all code and models in the paper.

Guidelines:

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

13. **New Assets**

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

    Answer: [Yes]

    Justification: We release a cleaned and simple-to use version of the model with working examples and an API. The code is linked at the end of the abstract.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We do not use human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [Yes]

Justification: We present a small-scale user study in our supplementary material, in which users were asked to rank explanations generated by our method and prior concept-based explanation methods. The study was ruled exempt by our institution's IRB, as no risks were posed to the study participants. Further information is given in Appendix B.1.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.