# SUPERVISED CONCEPT DISCOVERY FOR INTERPRETING VISION REPRESENTATIONS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Vision foundation models encode rich visual information, yet their representations do not directly expose human-interpretable factors. Concept discovery seeks to reveal these factors by decomposing pretrained representations into latent directions. However, reconstruction and sparsity objectives provide limited explicit guidance toward semantically coherent factors. We investigate class labels as a source of such guidance and introduce *label impurity*, an objective for guiding factor learning. Minimizing label impurity encourages each factor to exhibit consistent activation states among images of the same class, while allowing factors to be shared across classes. We derive an unbiased minibatch estimator for this objective to support stable stochastic optimization. Empirical results show that our approach discovers interpretable and visually coherent factors across diverse visual recognition datasets.

## 1 INTRODUCTION

Vision foundation models encode rich visual information that supports a wide range of downstream recognition tasks. Representations including this information, however, do not usually correspond to human-interpretable properties. Understanding these representations requires identifying the visual properties they encode, such as objects, textures, and attributes.

Recent methods often seek to linearly decompose pretrained embeddings into a set of concepts. That is, they represent a given embedding as a linear combination of concept vectors. Here, a concept denotes a human-interpretable factor. Related problems include identifying concept vectors first by human-annotations or large language models (LLMs) (Kim et al., 2018; Bhalla et al., 2024) and interpreting a given embedding through the corresponding concept vectors (Oikarinen & Weng, 2023)

A separate line of work, concept discovery, instead aims to discover potentially semantic factors directly from pretrained embedding without predefined concepts. Concept discovery can be viewed through the lens of dictionary learning, which learns a set of factors called a dictionary that collectively provide a compact description of the representations (Fel et al., 2023a; 2025). Methods for extracting the factors include clustering-based approaches (Ghorbani et al., 2019), matrix factorization methods (Zhang et al., 2021; Fel et al., 2023b), and sparse autoencoders (SAE) (Bricken et al., 2023; Huben et al., 2024). These methods can be understood as a reconstruction-based formulation, with different constraints on the dictionary and its coefficients. These constraints shape how representations are decomposed and which factors are recovered.

Among these, SAE learns an encoder that maps an image embedding to a sparse code. Each dimension of this code corresponds to a single (potentially semantic) factor, and its value indicates the extent to which the factor is activated for the input image. Recent studies demonstrate the effectiveness of SAE in visually coherent factors from pretrained vision representations (Rao et al., 2024; Lim et al., 2025; Pach et al., 2025). While these findings are encouraging, standard SAE objectives utilize no explicit supervision for learning semantically coherent factors and thus the extracting factors may not be fully aligned with semantic descriptions. Moreover, when there are task-specific semantic descriptions, SAE would not be successful to extract such factors.

The aim of this paper is to propose a new guidance for SAE-based concept discovery to learn factors better aligned with semantic interpretation. Building on the notion of *class specificity* observed by



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

Park et al. (2026), we investigate class membership as a source of guidance for concept discovery. Images within a same class often share common visual properties, some of which also occur across classes. Thus, we hypothesize that class membership provides a useful inductive bias for recovering semantically coherent factors, including those shared across classes.

Guided by this hypothesis, we introduce a *label impurity*, a specially designed supervised objective. Specifically, the label impurity encourages within-class activation consistency of learned factors. That is, factors which are activated or deactivated for all images in a same class are preferred. By adding the label impurity into the existing objective of SAE, we can improve the quality of semantic interpretation of learned factors. In other words, we are to infuse the known semantic meanings of class labels into learned factors.

To support stable optimization under the proposed objective, we develop an unbiased estimator of the label impurity which makes a mini-batch learning possible. In addition, we replace the ReLU activation function in SAE with the Heaviside activation function to learn binary factors This binary formulation of latent factors aligns naturally well with the within-class activation consistency encouraged by our label impurity objective and thus results in improved quality of semantic interpretation.

Our contributions are summarized as follows:

- We introduce a label impurity objective that uses class membership to guide SAE toward semantically coherent decompositions.
- We derive an unbiased estimator of the label impurity to support a mini-batch learning.
- We introduce an SAE with binary-valued activations that explicitly encode binary factors, providing better alignment with the proposed guidance and so semantic interpretation.
- Empirical studies show that the label impurity improves classification accuracy and semantic alignment, suggesting that our guidance helps learn concept sets better suited to the target task.

## 2 Related Works

### 2.1 Notation and Terminology

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ denote a labeled image dataset, where $\mathbf{x}_i \in \mathbb{R}^{H\times W\times C}$ and $y_i \in [K] := \{1, \dots, K\}$. We denote a random image–label pair by $(\mathbf{X}, Y)$ and define $\pi_k := \mathbb{P}(Y = k)$. A pretrained visual encoder $f$ maps each image to a representation $\mathbf{h}_i := f(\mathbf{x}_i) \in \mathbb{R}^p$.

Our aim is to learn a map $\phi : \mathbb{R}^p \to \mathcal{Z}^\ell$, where $\mathbf{z}_i = \phi(\mathbf{x}_i) \in \mathcal{Z}^\ell, i \in [N]$ are factors each of whose coordinates is semantically meaningful. We say that the $j$th factor is activated in the $i$th image if $z_{i,j} > 0$. In particular, we call $z_{i,j}$ a factor activation and $\mathbf{z}_i$ a *(factor) activation vector*. When $\mathbf{z}_i$ is sparse as in SAE, it is called a *sparse code*.

We use the term *concept discovery* for a process of recovering factors and examining their semantic interpretations. Here, *concepts* are factors having coherent sematic meanings. Concept discovery involves two stages: learning a factor map $\phi$ and extracting a semantic meaning of each factor. For the first stage, we modify SAE to use label information; for the second, we use an LLM or VLM to interpret the learned factors. See Sections 3.3 and 3.4 for details.

### 2.2 Sparse Autoencoders

Sparse Autoencoders (SAE) consist of an encoder that maps input representations to sparse codes and a decoder that reconstructs the input representations from these sparse codes (Bricken et al., 2023; Huben et al., 2024). Given a pretrained visual representation $\mathbf{h}_i \in \mathbb{R}^p$, the encoder produces a sparse code

$$\mathbf{z}_i^\theta := \phi_\theta(\mathbf{h}_i) = \rho\left(\boldsymbol{W}_{\text{enc}}^\top(\mathbf{h}_i - \mathbf{b})\right) \in \mathbb{R}^\ell,$$

where $\theta := (\boldsymbol{W}_{\text{enc}}, \mathbf{b})$ denotes the encoder parameter with $\boldsymbol{W}_{\text{enc}} \in \mathbb{R}^{p\times\ell}, \mathbf{b} \in \mathbb{R}^\ell$, and $\rho(\cdot)$ is a sparse non-linear activation function (e.g., ReLU). Note that the latent dimension $\ell$ is commonly much larger than the input dimension $p$, allowing SAE to learn overcomplete factors.

The decoder maps the sparse codes back to the original representations:

$$\widehat{\mathbf{h}}_i^{\theta,\xi} := \psi_\xi(\mathbf{z}_i) = \boldsymbol{W}_{\mathrm{dec}}^\top \mathbf{z}_i^\theta + \mathbf{b} \in \mathbb{R}^p,$$

for $\boldsymbol{W}_{\mathrm{dec}} \in \mathbb{R}^{\ell \times p}$, where $\xi := (\boldsymbol{W}_{\mathrm{dec}})$ denotes the decoder parameter.

ReLU-SAE, SAE with $\rho(\cdot) = \mathrm{ReLU}(\cdot)$, is trained by minimizing the sum of the reconstruction error and sparse penalty as

$$\mathcal{L}_{\mathrm{SAE}}(\theta, \xi) := \mathcal{L}_{\mathrm{rec}}(\theta, \xi) + \lambda_{\mathrm{sp}} \mathcal{L}_{\mathrm{sp}}(\theta),$$

where

$$\mathcal{L}_{\mathrm{rec}}(\theta, \xi) := \frac{1}{N} \sum_{i=1}^{N} \left\| \mathbf{h}_i - \widehat{\mathbf{h}}_i^{\theta,\xi} \right\|_2^2, \qquad \mathcal{L}_{\mathrm{sp}}(\theta) := \frac{1}{N} \sum_{i=1}^{N} \left\| \mathbf{z}_i^\theta \right\|_1, \tag{1}$$

and $\lambda_{\mathrm{sp}} > 0$ is a regularization parameter controlling the strength of the sparsity penalty. The reconstruction term encourages the decomposition to preserve information contained in the pretrained representations, while the sparsity penalty restricts the number and magnitude of factors that are simultaneously active for each representation.

### 2.3 Concept Discovery from Visual Representations

Various methods for extracting concepts differ along two dimensions: whether semantic identities are specified before or after extracting their representations, and whether information of class labels guides the extraction process or not. Table 1 summarizes this taxonomy and positions our method.

Table 1: Representative concept discovery strategies, organized by the order of semantic specification and the use of information of class labels in concept learning.

| **Discovery Strategies** | **Without class supervision** | **With class supervision** |
|---|---|---|
| **Concept-first** | Vocabulary-based decomposition<br>e.g., SpLiCE (Bhalla et al., 2024) | Class-conditioned vocabulary construction<br>e.g., LF-CBM (Oikarinen et al., 2023), LaBo (Yang et al., 2023) |
| **Factor-first** | Dictionary learning<br>e.g., CRAFT (Fel et al., 2023b), DN-CBM (Rao et al., 2024) | **Class-guided dictionary learning**<br>e.g., Ours |

**Concept-first extraction.** These methods represent images in terms of an explicitly specified concept vocabulary, but the vocabulary is constructed independently of a target classification task. For example, SpLiCE (Bhalla et al., 2024) first constructs a concept vocabulary by selecting frequent $n$-grams in LAION-400m (Schuhmann et al., 2021) captions. It then encodes these textual concepts using the CLIP text encoder to form a dictionary and decomposes image embeddings into sparse, non-negative combinations of the resulting dictionary vectors.

Alternatively, class information can guide vocabulary construction. Several concept-based image classification methods first use class labels to obtain candidate concepts using large language models, then construct a visual representation aligned with those concepts (Oikarinen et al., 2023; Yang et al., 2023; Srivastava et al., 2024). That is, the concept extraction stage employs a concept-first strategy : information of class labels determines *which concepts to represent* without looking at images, and visual representations are subsequently mapped to the specified set of concepts.

**Factor-first extraction.** These methods learn factors first without specifying their semantic identities. Representative approaches include Non-negative Matrix Factorization (NMF), as used in ICE and CRAFT (Zhang et al., 2021; Fel et al., 2023b), and SAE as used in DN-CBM (Rao et al., 2024). Their discovery mechanisms can be viewed as a dictionary learning approach (Fel et al., 2023a; 2025).

Given a collection of visual representations $\boldsymbol{A} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)]^\top \in \mathbb{R}^{N \times p}$, a common objective is

$$\min_{\boldsymbol{C}, \boldsymbol{D}} \|\boldsymbol{A} - \boldsymbol{C}\boldsymbol{D}\|_2^2 + \Omega(\boldsymbol{C}, \boldsymbol{D}),$$

where $\boldsymbol{C} = [\mathbf{z}_1, \ldots, \mathbf{z}_N]^\top \in \mathbb{R}^{N \times \ell}$ is a collection of activation vectors of images and $\boldsymbol{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_\ell]^\top \in \mathbb{R}^{\ell \times p}$ is the dictionary. The term $\Omega$ refers to regularizations or constraints on the decomposition. For example, NMF imposes non-negativity on the coefficients and dictionary, whereas SAE encourages the activation vectors to be sparse. These objectives seek to preserve information in the original representations while imposing certain structures on the recovered factors.

Our approach follows a factor-first strategy like SAE but incorporates information of class labels as an inductive bias.

### 2.4 Heavy Tailed Activation Function

The Heavy-Tailed Activation Function (HTAF) proposed by Park et al. (2026) is a surrogate of the Heavyside activation function $H(x) = \mathbb{I}(x \geq 0)$ for $x \in \mathbb{R}$, but is smooth so that gradient based optimization algorithms can be used for training. HTAF is defined as

$$\mathrm{HTAF}_{\alpha_0, \alpha_1}(x) = \sigma\left(\alpha_1 \tanh(\alpha_0 x)\right), \tag{2}$$

where $\sigma(x) = (1 + \exp(-x))^{-1}$ and $\alpha_0, \alpha_1 > 0$ are hyperparameters controlling the degree of approximation to the Heavyside activation function. During training, HTAF serves as a differentiable approximation to a binary activation, producing values near 0 or 1. At inference time, it is replaced back to the Heaviside activation function to obtain binary representations.

## 3 Method

Our model architecture is illustrated in Figure 1.

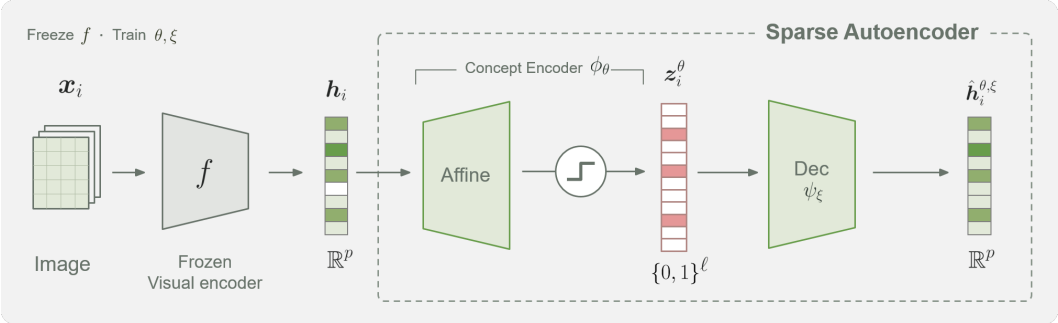



Figure 1: **Overview of the architecture.** Our model uses the same architecture as a standard SAE, except that it replaces the ReLU activation with a Heaviside activation.

### 3.1 Problem Setup

We instantiate $\phi_\theta$ and $\psi_\xi$ using SAE architecture introduced in Section 2.2, with the Heaviside function, $H(x) = \mathbb{I}(x \geq 0)$. This produces a binary activation vector $\mathbf{z}_i^\theta \in \{0,1\}^\ell$. During training, we replace the Heaviside function with HTAF, defined in (2), to enable gradient-based optimization.

### 3.2 Label Information as an Inductive Bias

For each class $k \in [K]$ and factor $j \in [\ell]$, we define the class-conditional mean activation as

$$\mu_{k,j}(\theta) = \frac{\sum_{i=1}^N z_{ij}^\theta \mathbb{I}(y_i = k)}{\sum_{i=1}^N \mathbb{I}(y_i = k)},$$

which is the conditional probability of activation of factor $j$ given an image belongs to class $k$. We introduce a form of semantic guidance that promotes class-wise consistency in the activation of each factor.

Specifically, we encourage $\mu_{k,j}(\theta)$ to approach either 0 or 1, promoting consistent inactivity or activity, respectively, of the $j$-th factor across images in class $k$. We quantify deviations from this property using the quadratic polarization function $q(u) = u(1-u)$. Minimizing $q(\mu_{k,j}(\theta))$ encourages the class-conditional mean activation to approach a binary extreme, which leads us to define the population *label impurity* as

$$\mathcal{R}_{\mathrm{LI}}(\theta) = \frac{1}{\ell} \sum_{j=1}^{\ell} \sum_{k=1}^{K} \pi_k q\left(\mu_{k,j}(\theta)\right). \tag{3}$$

**Remark.** One may argue that the standard cross-entropy (CE) could be used instead of the label impurity for supervision. The CE, however, prioritizes factors which are helpful for classification and thus learned factors could hamper the reconstruction error as well as sparsity as is shown in Figure 19. The poor sparsity may arise because the CE loss fails to identify useful concepts shared across multiple classes. The label impurity is carefully designed objective to transfer only the semantic meanings in the class labels to latent factors.

Computation of the proposed label impurity requires processing all training data, which hinders minibatch training. The following theorem, however, shows that there is an unbiased estimator of the label impurity from a given minibatch which is a degree-two $U$-statistic estimator (Hoeffding, 1948).

**Theorem 1** (Unbiased estimator of the label impurity)**.** *Let $\mathcal{B} := \{(\mathbf{x}_i, y_i)\}_{i=1}^{B}$ be a given minibatch that is a random sample from $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$.*

*Define the corresponding minibatch statistics for factor $j$,*

$$S_j^\theta := \sum_{i=1}^{B} z_{ij}^\theta, \qquad S_{k,j}^\theta := \sum_{i:i_i=k} z_{ij}^\theta, \qquad Q_{k,j}^\theta := \sum_{i:y_i=k} \left(z_{ij}^\theta\right)^2.$$

*Then,*

$$\widehat{\mathcal{R}}_{\mathrm{LI}}^{U}(\theta) := \frac{1}{\ell} \sum_{j=1}^{\ell} \left[ \frac{S_j^\theta}{B} - \frac{1}{B(B-1)} \sum_{k=1}^{K} \frac{(S_{k,j}^\theta)^2 - Q_{k,j}^\theta}{\pi_k} \right] \tag{4}$$

*is an unbiased estimator of the population objective in equation (3); that is,*

$$\mathbb{E}\left[\widehat{\mathcal{R}}_{\mathrm{LI}}^{U}(\theta)\right] = \mathcal{R}_{\mathrm{LI}}(\theta).$$

The class priors $\{\pi_k\}_{k=1}^{K}$ are computed from the full training dataset and treated as fixed during optimization. Because the estimator requires only class-wise sums and sums of squares, it can be evaluated in $O(B\ell)$ time. The proof of Theorem 1 is provided in Appendix B.

### 3.3 Training

We freeze the pretrained visual encoder $f$ and train SAE parameters $(\theta, \xi)$ on the target dataset.

Thus, we propose the following concept discovery objective:

$$\mathcal{L}_{\mathrm{ours}}(\theta, \xi) := \mathcal{L}_{\mathrm{rec}}(\theta, \xi) + \lambda_{\mathrm{sp}} \mathcal{L}_{\mathrm{sp}}(\theta) + \lambda_{\mathrm{LI}} \widehat{\mathcal{R}}_{\mathrm{LI}}^{U}(\theta), \tag{5}$$

where $\mathcal{L}_{\mathrm{rec}}$ and $\mathcal{L}_{\mathrm{sp}}$ are defined as (1). Here, $\lambda_{\mathrm{sp}} > 0$ and $\lambda_{\mathrm{LI}} > 0$ control the strengths of sparsity and label impurity regularization, respectively. Plus, the Heaviside activation function is replaced by HTAF during the training.

Given objective (5) includes the sparsity penalty, even the activation vectors are binary. It is interesting to see that the $\ell_1$ penalty for binary factors results in the $\ell_0$ penalty. That is, when $\mathbf{z}_i^\theta$ is binary, then

$$\|\mathbf{z}_i^\theta\|_1 = \sum_{j=1}^{\ell} \mathbb{I}(z_{i,j}^\theta = 1) = \sum_{j=1}^{\ell} \mathbb{I}(z_{i,j}^\theta \neq 0).$$

Hence, the $\ell_1$ penalty still plays an important role for controlling the sparsity.

### 3.4 Extracting Semantic Meanings

After training SAE, we assign a name to each factor using one of two strategies, depending on the vocabulary coverage required by the target dataset.

**Decoder-based naming for general object and scene recognition.** We adopt the vocabulary-based naming scheme of DN-CBM (Rao et al., 2024), using a predefined vocabulary of common visual concepts. Suppose that pretrained representations are CLIP embedding vectors of each image. Then, each factor is named by matching its decoder direction to a text embedding in CLIP representation space. Formally, let $\mathcal{V}$ be the vocabulary set and $\mathrm{Enc}_{\mathrm{text}}$ denote the CLIP text encoder. The decoder direction associated with factor $j$ is

$$\boldsymbol{W}_{\mathrm{dec},j} := (\boldsymbol{W}_{\mathrm{dec}})_{j,:} \in \mathbb{R}^p.$$

We name factor $j$ by selecting the vocabulary whose text embedding has the highest cosine similarity to its decoder direction:

$$\tau_j := \arg\max_{\tau \in \mathcal{V}} \cos\left(\boldsymbol{W}_{\mathrm{dec},j}, \mathrm{Enc}_{\mathrm{text}}(\tau)\right), \tag{6}$$

where $\cos(\cdot,\cdot)$ denotes cosine similarity. We refer to the similarity $\cos\left(\boldsymbol{W}_{\mathrm{dec},j}, \mathrm{Enc}_{\mathrm{text}}(\tau_j)\right)$ as the *naming similarity* of factor $j$.

Alternatively, we consider letting the direction corresponding to each factor as the average of CLIP visual embedding of the images for which that factor is active. The two approaches are equivalent when the factors are orthogonal. Unlike the first approach, however, this alternative can also be applied to pretrained representations outside the CLIP embedding space.

**LLM-based naming for fine-grained recognition.** For fine-grained datasets, such as CUB-200 (Wah et al., 2011), a general-purpose vocabulary may lack the domain-specific attributes needed for describing the discovered factors. When an appropriate term is absent, decoder-based matching selects vocabulary, potentially assigning a misleading name even to a given factor.

We therefore use an LLM to generate factor names from the class labels of their top-activating images, rather than retrieving names from a fixed vocabulary. Specifically, for each factor, we collect the labels of classes whose images are consistently activated and prompt the LLM to propose a shared visual characteristic using its domain knowledge of the corresponding categories.

## 4 Experiments

### 4.1 Baselines and Datasets

**Baslines.** As a baseline, we use SAE pretrained on CC3M by Rao et al. (2024). CC3M is a large-scale image–caption dataset (Sharma et al., 2018) that contains a diverse range of images. Therefore, the pretrained SAE is expected to capture task-agnostic concepts.

We also train an SAE with the architecture used in DN-CBM (Rao et al., 2024) separately for each dataset. We refer to this model as **SAE** in this section. In addition, we consider the following two variants:

- **SAE+LI**: an SAE trained with the objective in (5), which includes the label-impurity loss.
- **SAE+Heaviside+LI** (Ours): an SAE+LI model in which the ReLU activation is replaced with a Heaviside activation.

**Datasets.** For a comprehensive evaluation, we conduct experiments on two broad-coverage datasets including ImageNet-1K (Deng et al., 2009) and Places365 (Zhou et al., 2018), as well as the fine-grained dataset CUB-200 (Wah et al., 2011).

Table 2: **Datasets Descriptions.**

| | Fine-grained | # Classes | Class label examples |
|---|---|---|---|
| ImageNet-1K | ✗ | 1000 | Golden Retriever, Sports car, Mushroom |
| CUB-200 | ✓ | 200 | Black-footed Albatross, Mallard, Blue Jay |
| Places365 | ✗ | 365 | Airport terminal, Beach, Volcano |

All the analyses including Sections 4.2.2 and 4.3 use the parameter settings that yield the highest test accuracy.

## 4.2 Quantitative Performances

### 4.2.1 Classification Accuracy

Table 3 reports the accuracies of the linear classifiers built on the factors learned by SAE, SAE+LI and SAE+HTAF+LI as well as DN-CBM. In addition, the linear classifier trained on the pretrained representations is also considered (Linear Probe). The results show that adding the label impurity improves overall classification performance while replacing ReLU by HTAF is not helpful for prediction accuracy.

It is interesting to see that DN-CBM is more accurate than SAE even though SAE is trained on data related to a given task while DN-CBM is task agnostic. A possible reason is that the data set used for training DN-CBM is much larger and more diverse. In contrast, SAE+LI and SAE+Heaviside+LI use the information of class labels to make improvement of prediction accuracy. However, in Section 4.3.2, we illustrate that learned concepts of our methods from one data could be transferred to useful concepts for another data even though our methods are not task agnostic. This empirically shows that our methods do not utilize label information too much.

### 4.2.2 Quality of Semantic Interpretation

In this section, we assess whether the discovered and named concepts are semantically coherent. The evaluation procedure is illustrated in Figure 2. We sample a single image from each class, and obtain the cosine similarity between an image CLIP embedding and average of textual concept CLIP embeddings of activated concepts.

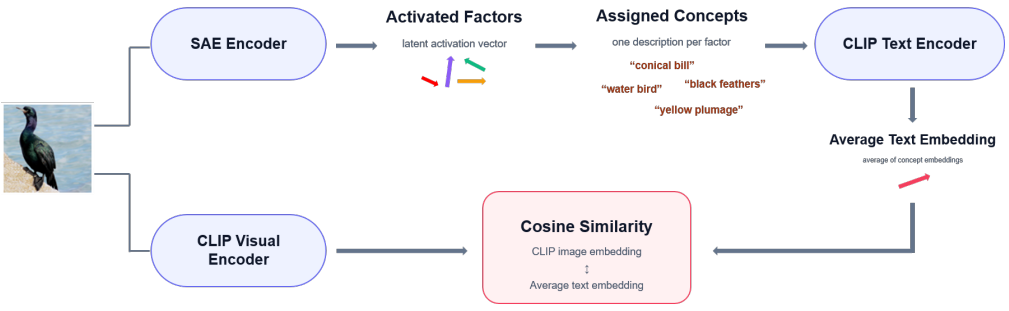



Figure 2: **Semantic alignment evaluation procedure.**

The results in Table 4 are the averages over the cosine similarities between each image and each of top $K$ concepts, which indicates that the label impurity and binary concepts are helpful for improving the sematic alignment. The factors are ranked and selected based on the importance for prediction of class labels.

Table 3: **Test accuracy on ImageNet, CUB-200, and Places365.** For DN-CBM, we used the released pretrained SAE and re-evaluated on each down-stream task.

| | CLIP-ViT-B/16 | | | CLIP-ViT-L/14 | | |
|---|---|---|---|---|---|---|
| Method | ImageNet | CUB-200 | Places365 | ImageNet | CUB-200 | Places365 |
| Linear Probe | 0.800 | 0.808 | 0.552 | 0.840 | 0.860 | 0.556 |
| DN-CBM (Rao et al., 2024) | 0.789 | 0.795 | 0.550 | 0.815 | 0.857 | 0.551 |
| SAE | 0.736 | 0.783 | 0.501 | 0.784 | 0.841 | 0.548 |
| SAE + LI | **0.796** | **0.795** | **0.549** | **0.842** | **0.860** | **0.557** |
| SAE+HTAF+LI | 0.765 | 0.766 | 0.529 | 0.833 | 0.854 | 0.544 |

Table 4: **Cosine similarities between averaged text embeddings and image embeddings.**

| Dataset | # Factors | SAE | SAE+LI | SAE+HTAF+LI | DN-CBM |
|---|---|---|---|---|---|
| ImageNet | 10 | 0.238 | 0.245 | **0.261** | 0.245 |
| | 20 | 0.235 | 0.245 | **0.261** | 0.244 |
| | 30 | 0.230 | 0.242 | **0.259** | 0.238 |
| | 40 | 0.229 | 0.240 | **0.259** | 0.237 |
| CUB-200 | 10 | 0.282 | 0.292 | **0.306** | 0.251 |
| | 20 | 0.284 | 0.296 | **0.306** | 0.244 |
| | 30 | 0.285 | 0.297 | **0.309** | 0.241 |
| | 40 | 0.282 | 0.299 | **0.308** | 0.237 |

#### 4.2.3 Sparsity Analysis

We examine the trade-offs among representational sparsity, test accuracy, and reconstruction loss on CUB-200. We measure sparsity by the number of alive factors: an SAE factor is considered dead if it has zero activation for the majority of images, whereas a binary factor is considered dead if it remains constant at either 0 or 1 across the dataset.

Figure 3 shows that the label-impurity loss reduces the number of alive factors while maintaining test accuracy, and the Heaviside activation yields an even more compact representation with comparable predictive performance. These results indicate that our method captures task-relevant information using fewer factors. Although this increased sparsity is accompanied by higher reconstruction loss, the preserved test accuracy suggests that the learned representation may selectively retain information relevant to CUB-200 classification rather than reconstructing all information in the pretrained embeddings.

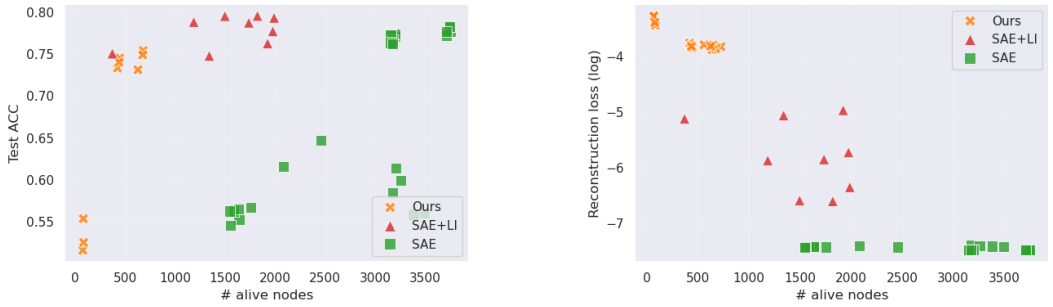



Figure 3: **Sparsity–accuracy and sparsity–reconstruction trade-offs.** We compare our method, SAE+LI and SAE across different levels of sparsity, measured by the number of alive nodes. The models are all trained on CUB-200 dataset with CLIP ViT-B/16.

### 4.3 Qualitative Evaluations

#### 4.3.1 Concepts Discovered on Each Dataset

We first qualitatively inspect the concepts discovered by SAE trained on each dataset. Figure 4 presents several representative concepts together with their top-activating images. For datasets that use decoder-based naming, we display the concept with the highest naming similarity. For CUB-200, we display the factor with the lowest factor-wise label impurity, i.e., $\sum_{k=1}^{K} q(\mu_{k,j}(\theta))$.

These examples suggest that the proposed framework can discover factors corresponding to human-interpretable and visually coherent concepts. Additional results are shown in Appendix C.

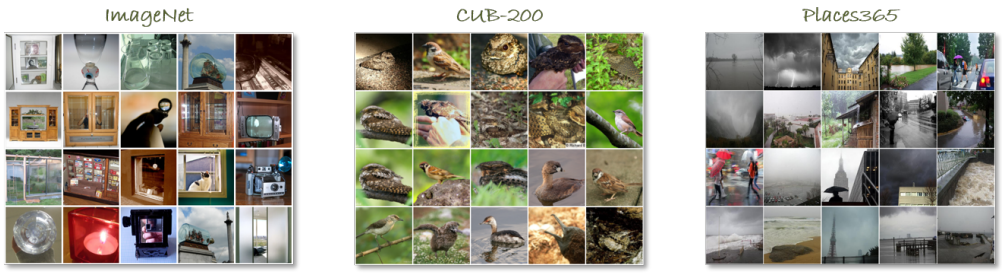



Figure 4: **Representative concepts discovered on each dataset.** Each panel shows the top-20 activating images for a representative factor learned from the corresponding dataset. From left to right, the discovered concepts are named *window*, *brownish plumage*, and *rainy*.

#### 4.3.2 Cross-Dataset Concept Coherence

In this section, we investigate whether concepts trained on one dataset can be transferred to other datasets. Specifically, we brought our SAE which was trained on ImageNet and calculate the sparse codes of the images from the other datasets, CUB-200 and Places365. Then, we visually check the validity of each concept, that is, determine whether the named concepts remain semantically coherent across datasets. The results are shown in Figure 5. Additional results are provided in Appendix C.




Figure 5: **Top-activating images for ImageNet-trained SAE factors across datasets.** For each named concept under ImageNet, we show the top-20 activating images from CUB-200 and Places 365, ranked by their pre-activation values.

## 5 Concluding Remarks

In this work, we introduced a framework for discovering sparse, task-specific concepts from pre-trained visual representations. Our empirical results show that the learned factors capture visually coherent concepts while retaining information relevant to downstream prediction.

Several directions for future work remain. First, binary classification requires a new objective, as the current label impurity formulation applies only to multiclass classification. Second, the framework could be extended beyond class-labeled datasets by leveraging the richer semantic information available in image-caption datasets. Lastly, incorporating a gated SAE (Rajamanoharan et al., 2024) with the label impurity and Heaviside activation would improve predictive performance and reconstruction quality while preserving semantic alignment.

### AI USE STATEMENT

In this work, we used generative AI tools for to aid and polish writing. In addition, we used LLM which is incorporated in our proposed framework, and its elaboration is included in the manuscript. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, code, and artifacts produced with the aid of generative AI.

### REPRODUCIBILITY STATEMENT

We provide the information as much as we can including the checkpoints url, hyperparameter settings, and data split configurations. Details are provided in Appendix A.

## REFERENCES

Usha Bhalla, Alex Oesterling, Suraj Srinivas, Flavio P Calmon, and Himabindu Lakkaraju. Interpreting clip with sparse linear concept embeddings (splice). *Advances in Neural Information Processing Systems*, 37:84298–84328, 2024.

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

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. IEEE, 2009.

Thomas Fel, Victor Boutin, Louis Béthune, Rémi Cadène, Mazda Moayeri, Léo Andéol, Mathieu Chalvidal, and Thomas Serre. A holistic approach to unifying automatic concept extraction and concept importance estimation. *Advances in Neural Information Processing Systems*, 36:54805–54818, 2023a.

Thomas Fel, Agustin Picard, Louis Bethune, Thibaut Boissin, David Vigouroux, Julien Colin, Rémi Cadène, and Thomas Serre. Craft: Concept recursive activation factorization for explainability. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2711–2721, 2023b.

Thomas Fel, Ekdeep Singh Lubana, Jacob S Prince, Matthew Kowal, Victor Boutin, Isabel Papadimitriou, Binxu Wang, Martin Wattenberg, Demba Ba, and Talia Konkle. Archetypal sae: Adaptive and stable dictionary learning for concept extraction in large vision models. *arXiv preprint arXiv:2502.12892*, 2025.

Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 3681–3688, 2019.

Wassily Hoeffding. A class of statistics with asymptotically normal distribution. *The Annals of Mathematical Statistics*, 19(3):293–325, 1948. doi: 10.1214/aoms/1177730196.

Robert Huben, Hoagy Cunningham, Logan Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations*, volume 2024, pp. 7827–7845, 2024.

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*, pp. 2668–2677. PMLR, 2018.

Hyesu Lim, Jinho Choi, Jaegul Choo, and Steffen Schneider. Sparse autoencoders reveal selective remapping of visual concepts during adaptation. In *International Conference on Learning Representations*, volume 2025, pp. 24444–24469, 2025.

Tuomas Oikarinen and Tsui-Wei Weng. CLIP-dissect: Automatic description of neuron representations in deep vision networks. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=iPWiwWHc1V.

Tuomas Oikarinen, Subhro Das, Lam M. Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=FlCg47MNvBA.

Mateusz Pach, Shyamgopal Karthik, Quentin Bouniot, Serge Belongie, and Zeynep Akata. Sparse autoencoders learn monosemantic features in vision-language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=DaNnkQJSQf.

Seokhun Park, Choeun Kim, Kwanho Lee, Sehyun Park, Insung Kong, and Yongdai Kim. A composite activation function for learning stable binary representations. *arXiv preprint arXiv:2605.11558*, 2026.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah, and Neel Nanda. Improving sparse decomposition of language model activations with gated sparse autoencoders. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 775–818. Curran Associates, Inc., 2024. doi: 10.52202/079017-0024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/01772a8b0420baec00c4d59fe2fbace6-Paper-Conference.pdf.

Sukrut Rao, Sweta Mahajan, Moritz Böhle, and Bernt Schiele. Discover-then-name: Task-agnostic concept bottlenecks via automated concept discovery. In *European Conference on Computer Vision*, pp. 444–461. Springer, 2024.

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.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2556–2565, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1238. URL https://aclanthology.org/P18-1238/.

Divyansh Srivastava, Ge Yan, and Tsui-Wei Weng. Vlg-cbm: Training concept bottleneck models with vision-language guidance. *Advances in Neural Information Processing Systems*, 37:79057–79094, 2024.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19187–19197. IEEE, 2023.

Ruihan Zhang, Prashan Madumal, Tim Miller, Krista A Ehinger, and Benjamin IP Rubinstein. Invertible concept-based explanations for cnn models with non-negative concept activation vectors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11682–11690, 2021.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018. doi: 10.1109/TPAMI.2017.2723009.

# A Experimental Details

**CLIP checkpoints.** We used the pretrained CLIP-ViT-B/16 and CLIP-ViT-L/14 visual encoders. The corresponding checkpoints, `openai/clip-vit-base-patch16` and `openai/clip-vit-large-patch14`, were loaded using the Hugging Face `transformers` library. The visual encoders were frozen throughout SAE training and linear probing.

**DN-CBM SAE checkpoints.** For our re-evaluation of DN-CBM, we used the publicly released SAE checkpoint trained on CC3M.[1]

**Concept-naming vocabulary.** We used a publicly available vocabulary of 20,000 English words as the candidate set for concept naming.[2] Following Equation (6), each SAE node was assigned the vocabulary term whose CLIP text embedding had the highest cosine similarity to the node's decoder direction.

**Hyperparameter settings.** We tuned the weights of class-specificity regularization and the $\ell_1$ sparsity penalty over the search spaces reported in Table 5.

Table 5: **Hyperparameter search spaces for our method**

| Hyperparameter | Candidate values |
|---|---|
| ReLU SAE Label Impurity weight $\lambda_{\mathrm{LI}}$ | $\{1,\ 3,\ 10\}$ |
| ReLU SAE Sparsity weight $\lambda_{\mathrm{sp}}$ | $\{3e-5,\ 1e-4,\ 3e-4\}$ |
| Label Impurity weight $\lambda_{\mathrm{LI}}$ (ours) | $\{0.5,\ 1,\ 5\}$ |
| Sparsity weight $\lambda_{\mathrm{sp}}$ (ours) | $\{2e-5,\ 3e-5,\ 4e-5\}$ |
| SAE learning rate | $\{1e-4,\ 3e-4,\ 5e-4\}$ |
| SAE epochs | 500 for CUB-200, 50 otherwise |
| SAE expansion factor | 8 |
| Linear probing learning rate | $\{1e-4,\ 3e-4\}$ |
| Linear probing epochs | 500 for CUB-200, 50 otherwise |
| HTAF $(\alpha_0, \alpha_1)$ | (3, 9) |

The setting $\lambda_{\mathrm{LI}} = 0$ corresponds to the ablation without class-specificity regularization. All other training settings, including the selected sparsity weight, were kept identical when constructing this ablation.

**Dataset Splits and Feature discovery.** For all datasets, we randomly split the designated training data into training and validation subsets with a 99:1 ratio, using seed 42. We extract and cache CLIP image features and their corresponding labels for every split before training.

For ImageNet (Deng et al., 2009), we split the official training set into training and validation subsets. We use the official validation set as the test set, as labels for the official test set are not publicly available.

For Places365 (Zhou et al., 2018), we use the `train-standard` split provided by `torchvision.datasets` as the source of the training and validation data. We use the provided validation split as the test set.

For CUB-200 (Wah et al., 2011), we split the official training set into training and validation subsets and use the official test split for evaluation.

[1] `https://nextcloud.mpi-klsb.mpg.de/index.php/s/PZYoKSq29XaEgL6`

[2] `https://github.com/first20hours/google-10000-english/blob/master/20k.txt`

## B Derivation of the Label Impurity Loss

A naive plug-in estimator replaces $\mu_{k,j}(\theta)$ in equation (3) with an empirical conditional mean $\widehat{\mu}_{k,j}$. Since $q(\cdot)$ is nonlinear, we generally have

$$\mathbb{E}\left[q\left(\widehat{\mu}_{k,j}\right)\right] \neq q\left(\mu_{k,j}(\theta)\right). \tag{7}$$

Let

$$\mathbf{V}_1 := (\mathbf{X}_1, Y_1), \qquad \mathbf{V}_2 := (\mathbf{X}_2, Y_2)$$

be i.i.d copies of $(\mathbf{X}, Y)$.

Define

$$Z_{1j}^{\theta} := \left[\phi_\theta(\mathbf{X}_1)\right]_j, \qquad Z_{2j}^{\theta} := \left[\phi_\theta(\mathbf{X}_2)\right]_j,$$

and the symmetric kernel

$$h_{\theta,j}(\mathbf{V}_1, \mathbf{V}_2) := \frac{Z_{1j}^{\theta} + Z_{2j}^{\theta}}{2} - \frac{\mathbb{I}(Y_1 = Y_2)}{\pi_{Y_1}} Z_{1j}^{\theta} Z_{2j}^{\theta}.$$

The expectation of the first term is

$$\begin{aligned}
\mathbb{E}\left[\frac{Z_{1j}^{\theta} + Z_{2j}^{\theta}}{2}\right] &= \mathbb{E}\left[Z_{1j}^{\theta}\right] \\
&= \mathbb{E}_{Y_1}\left[\mathbb{E}\left[Z_{1j}^{\theta} \mid Y_1\right]\right] \\
&= \sum_{k=1}^{K} \pi_k \mathbb{E}\left[Z_{1j}^{\theta} \mid Y_1 = k\right] \\
&= \sum_{k=1}^{K} \pi_k \mu_{k,j}(\theta).
\end{aligned} \tag{8}$$

For the second term, the law of total expectation and the independence of the two samples give

$$\begin{aligned}
\mathbb{E}\left[\frac{\mathbb{I}(Y_1 = Y_2)}{\pi_{Y_1}} Z_{1j}^{\theta} Z_{2j}^{\theta}\right] &= \mathbb{E}_{Y_1, Y_2}\left[\frac{\mathbb{I}(Y_1 = Y_2)}{\pi_{Y_1}} \mathbb{E}\left[Z_{1j}^{\theta} Z_{2j}^{\theta} \mid Y_1, Y_2\right]\right] \\
&= \sum_{k=1}^{K} \frac{\mathbb{P}(Y_1 = k, Y_2 = k)}{\pi_k} \mathbb{E}\left[Z_{1j}^{\theta} Z_{2j}^{\theta} \mid Y_1 = k, Y_2 = k\right] \\
&= \sum_{k=1}^{K} \frac{\pi_k^2}{\pi_k} \mathbb{E}\left[Z_{1j}^{\theta} \mid Y_1 = k\right] \mathbb{E}\left[Z_{2j}^{\theta} \mid Y_2 = k\right] \\
&= \sum_{k=1}^{K} \pi_k \mu_{k,j}^2(\theta).
\end{aligned} \tag{9}$$

From (8) and (9), we have

$$\mathbb{E}\left[h_{\theta,j}(\mathbf{V}_1, \mathbf{V}_2)\right] = \sum_{k=1}^{K} \pi_k \mu_{k,j}(\theta)\left\{1 - \mu_{k,j}(\theta)\right\}. \tag{10}$$

For an i.i.d. minibatch $\mathcal{B} = \{(\mathbf{X}_i, Y_i)\}_{i=1}^{B}$, define the complete degree-two $U$-statistic as

$$\widehat{\mathcal{R}}_{\mathrm{LI},j}^{U}(\theta) := \binom{B}{2}^{-1} \sum_{1 \leq i < i' \leq B} h_{\theta,j}(\mathbf{V}_i, \mathbf{V}_{i'}), \tag{11}$$

and

$$S_j^\theta := \sum_{i=1}^{B} Z_{ij}^\theta, \qquad S_{k,j}^\theta := \sum_{i:Y_i=k} Z_{ij}^\theta, \qquad Q_{k,j}^\theta := \sum_{i:Y_i=k} \left(Z_{ij}^\theta\right)^2. \tag{12}$$

By equation (10),

$$\mathbb{E}\left[\widehat{\mathcal{R}}_{\mathrm{LI},j}^{U}(\theta)\right] = \sum_{k=1}^{K} \pi_k q\left(\mu_{k,j}(\theta)\right). \tag{13}$$

Although the estimator in equation (11) appears to require $O(B^2)$ pairwise operations, its quadratic form permits a linear-time implementation.

For the first term of the kernel, each activation appears in exactly $B-1$ pairs, and hence

$$\binom{B}{2}^{-1} \sum_{1\le i<i'\le B} \frac{Z_{ij}^\theta + Z_{i'j}^\theta}{2} = \frac{S_j}{B}. \tag{14}$$

For each class $k$, we have

$$\begin{aligned}
\binom{B}{2}^{-1} \sum_{1\le i\le i'\le B} \frac{\mathbb{I}(Y_i = Y_{i'})}{\pi_{Y_i}} Z_{ij}^\theta Z_{i'j}^\theta &= \binom{B}{2}^{-1} \sum_{1\le i<i'\le B} \sum_{k=1}^{K} \frac{\mathbb{I}(Y_i = Y_i' = k)}{\pi_{Y_i}} Z_{ij}^\theta Z_{i'j}^\theta \\
&= \binom{B}{2}^{-1} \sum_{1\le i<i'\le B} \sum_{k=1}^{K} \frac{\mathbb{I}(Y_i = Y_i' = k)}{\pi_{k}} Z_{ij}^\theta Z_{i'j}^\theta \\
&= \binom{B}{2}^{-1} \sum_{k=1}^{K} \sum_{1\le i<i'\le B} \frac{\mathbb{I}(Y_i = Y_i' = k)}{\pi_{k}} Z_{ij}^\theta Z_{i'j}^\theta \\
&= \binom{B}{2}^{-1} \sum_{k=1}^{K} \frac{1}{2} \frac{(S_{k,j}^\theta)^2 - Q_{k,j}^\theta}{\pi_k} \\
&= \frac{1}{B(B-1)} \sum_{k=1}^{K} \left(S_{k,j}^2 - Q_{k,j}\right).
\end{aligned} \tag{15}$$

Substituting equations (14) and (15) into equation (11) gives

$$\widehat{\mathcal{R}}_{\mathrm{LI},j}^{U}(\theta) = \frac{S_j}{B} - \frac{1}{B(B-1)} \sum_{k=1}^{K} \frac{S_{k,j}^2 - Q_{k,j}}{\pi_k}. \tag{16}$$

Averaging equation (16) over $j \in [\ell]$ yields equation (4). Consequently, the complete regularizer can be evaluated in $O(Bd)$ time without explicitly constructing all pairs.

# C Additional Results for Qualitative Evaluation on Concept Discovery

**Concepts Discovered on Each Dataset.** We present additional results on ImageNet using CLIP ViT-B/16. The panels show results for the factors with the highest naming similarity.

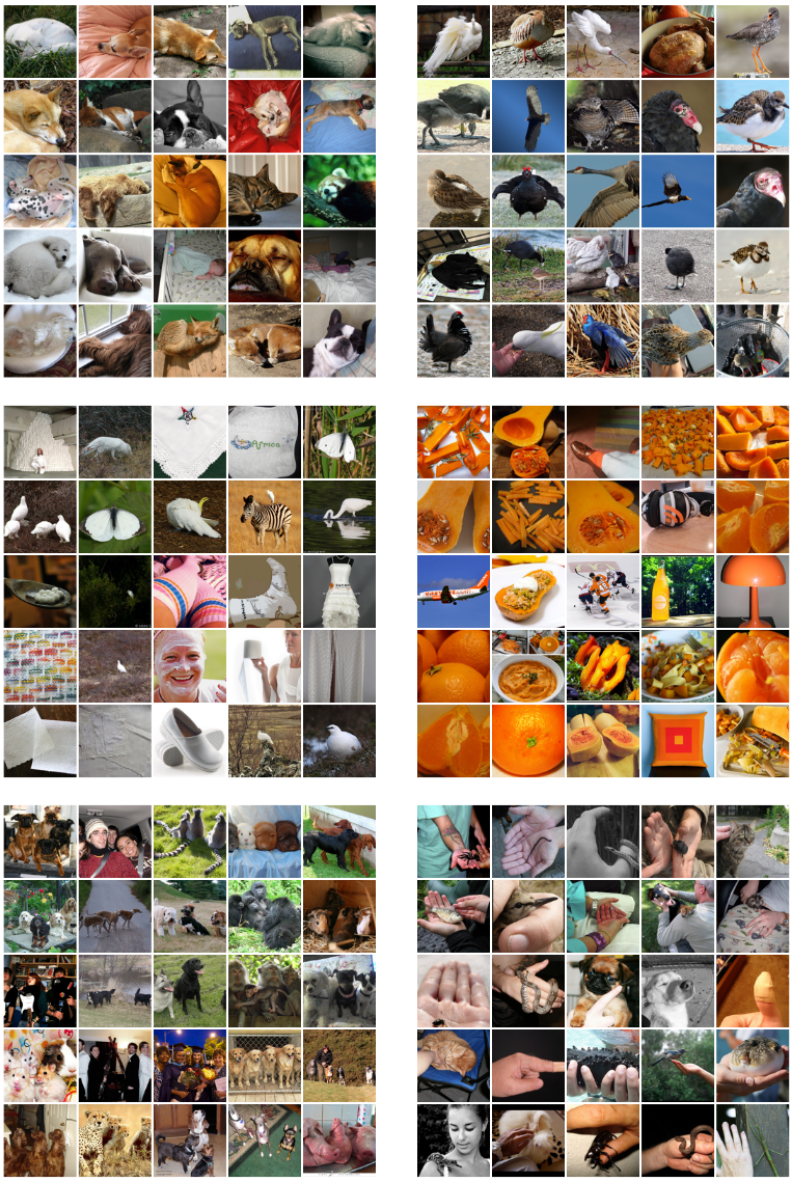


Figure 6: **Concepts :** sleeping, birds, white, orange, quartet, handful from left to right, top to bottom.

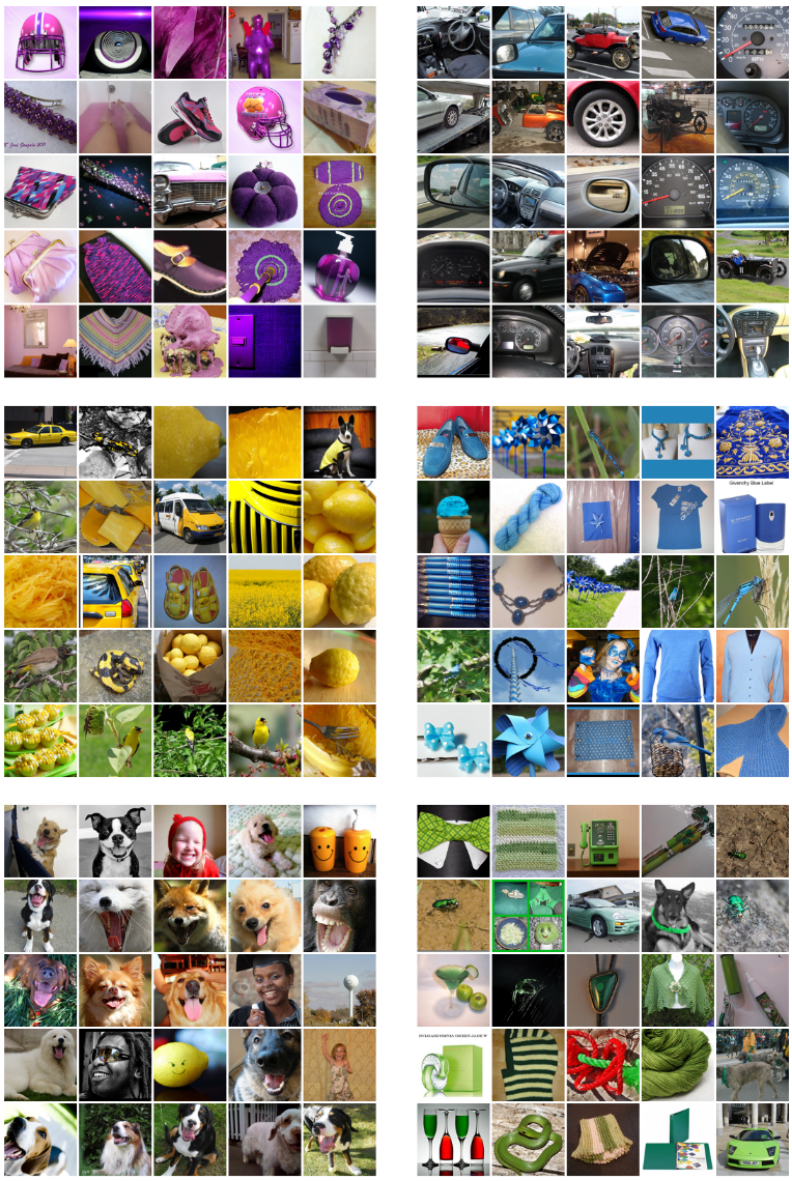


Figure 7: **Concepts :** purple, car, yellow, blue, smilies, green from left to right, top to bottom.

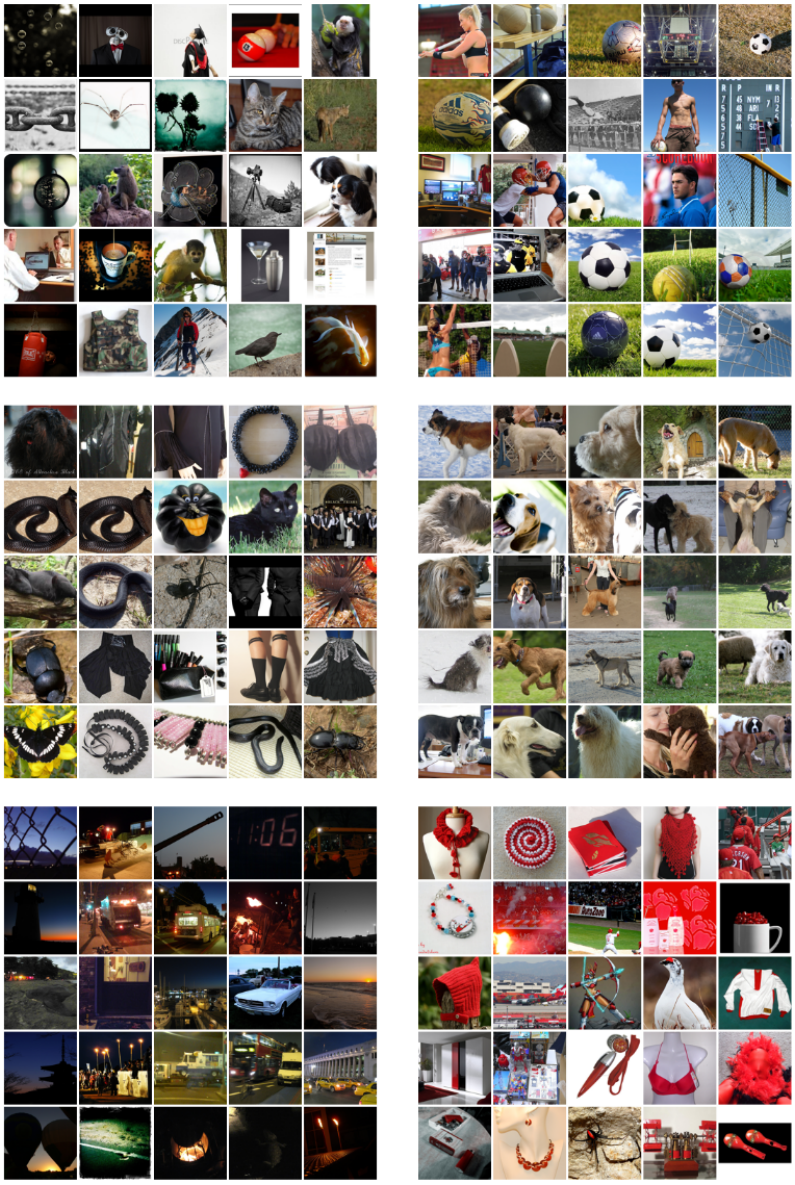


Figure 8: **Concepts:** social, softball, black, dog, dusk, red from left to right, top to bottom.

**Cross-data Concept Analysis.** Additional results on Section 4.3.2. All the panels are from ImageNet, CUB-200, and Places365 from left to right.



Figure 9: Dogs



Figure 10: Appetite



Figure 11: Beverage

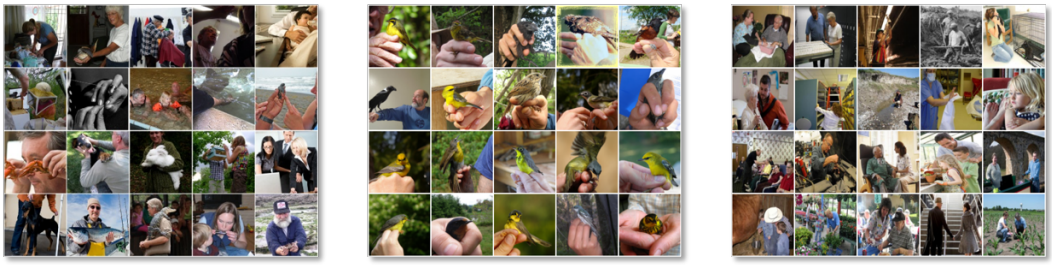


Figure 12: Handful



Figure 13: Ocean



Figure 14: Pond



Figure 15: Pink



Figure 16: Orange



Figure 17: Otter



Figure 18: White

## D Comparison with Cross-Entropy Loss

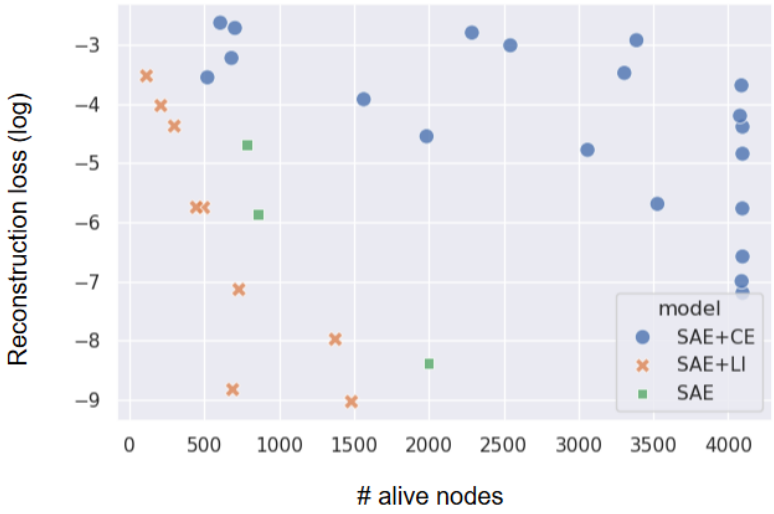



Figure 19: **Sparsity-reconstruction trade-off on ImageNet with CLIP Vit-B/16.** Each point denotes one trained model configuration. SAE with label impurity loss achieves substantially lower reconstruction loss with fewer live nodes than SAE with cross entropy loss.

Another approach for incorporating class information into SAE training is to add a linear classifier on top of the sparse code and train them in end-to-end manner with the standard SAE objective and a cross-entropy loss. The resulting objective is

$$\mathcal{L}_{\text{E2E}} = \mathcal{L}_{\text{CE}} + \lambda_{\text{rec}}\mathcal{L}_{\text{recon}} + \lambda_{\text{sp}}\mathcal{L}_{\text{sp}}. \tag{17}$$

We examined whether the cross-entropy loss yields a useful trade-off between sparsity and reconstruction quality. We set $\lambda_{\text{rec}} \in \{5, 10, 50, 100\}$, and $\lambda_{\text{sp}} \in \{10^{-3}, 3\times10^{-3}, 10^{-2}, 3\times10^{-2}, 10^{-1}\}$.

Figure 19 shows that cross-entropy supervision does not consistently yield a favorable sparsity–reconstruction trade-off: configurations with fewer live nodes generally incur higher reconstruction loss than ours. In contrast, label impurity objective achieves lower reconstruction loss at a comparable level of sparsity, suggesting that it provides a more effective way to incorporate label information while preserving sparse, accurate decompositions.