Hierarchical Concept Embedding & Pursuit for Interpretable Image Classification

Nghia Nguyen Tianjiao Ding René Vidal
University of Pennsylvania
nghianhh@seas.upenn.edu

Abstract

Interpretable-by-design models are gaining traction in computer vision because they provide faithful explanations for their predictions. In image classification, these models typically recover human-interpretable concepts from an image and use them for classification. Sparse concept recovery methods leverage the latent space of vision-language models to represent image embeddings as sparse combinations of concept embeddings. However, by ignoring the hierarchical structure of semantic concepts, these methods may produce correct predictions with explanations that are inconsistent with the hierarchy. In this work, we propose Hierarchical Concept Embedding & Pursuit (HCEP), a framework that induces a hierarchy of concept embeddings in the latent space and performs hierarchical sparse coding to recover the concepts present in an image. Given a hierarchy of semantic concepts, we introduce a geometric construction for the corresponding hierarchy of embeddings. Under the assumption that the true concepts form a rooted path in the hierarchy, we derive sufficient conditions for their recovery in the embedding space. We further show that hierarchical sparse coding reliably recovers hierarchical concept embeddings, whereas standard sparse coding fails. Experiments on real-world datasets show that HCEP improves concept precision and recall compared to existing methods while maintaining competitive classification accuracy. Moreover, when the number of samples available for concept estimation and classifier training is limited, HCEP achieves superior classification accuracy and concept recovery. Our results demonstrate that incorporating hierarchical structure into sparse concept recovery leads to more faithful and interpretable image classification models.[1]

1. Introduction

Machine learning has been adopted in many computer vision applications, including image classification, question answering, and concept recovery, often matching or surpassing human performance [33, 55]. However, the lack of interpretability in these models' predictions has raised concerns about their trustworthiness [16, 40, 57].

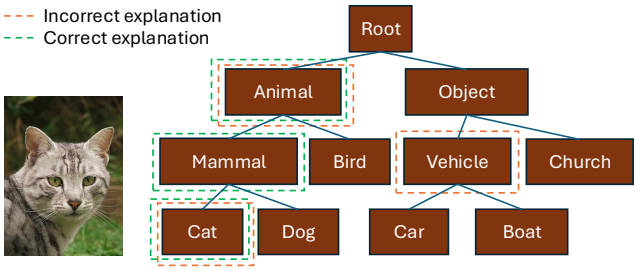

Figure 1. **An illustration of hierarchical concept explanations for image classification.** Given an image of a `cat`, an interpretable-by-design model should extract the concepts that form a rooted path in the hierarchy, and use only these concepts to classify the image as a `cat`. However, existing concept recovery methods may extract concepts that are inconsistent with the hierarchy, such as `vehicle`, leading to incorrect explanations.

Interpretable models in computer vision generally fall into two categories: *post-hoc explanation* and *interpretable-by-design* methods. Post-hoc explanation methods aim to provide insights into the decision-making process of trained black-box models [42, 56, 60, 62]. However, such methods often suffer from a lack of faithfulness and stability with respect to the original pre-trained models [1, 5]. This paper focuses on interpretable-by-design methods, which build interpretability directly into the model training process [2, 10, 29]. Such models usually consist of two steps: (1) extracting human-interpretable concepts from the input and (2) using only these concepts for downstream tasks such as classification or regression.[2] For instance, given an image of a cat, an interpretable-by-design model would extract concepts such as `animal`, `furry`, and `short muzzle`, and then classify the image using only these concepts.

Recent methods for recovering human-interpretable concepts from images differ in how they use supervision and whether they can handle unseen concepts. In *fully supervised concept recovery*, a model is trained to predict a pre-

[1] Code at https://github.com/nghiahhnguyen/hcep.

[2] There is an emerging use of Chain-of-Thought [68] as an interpretability method, which is not fully interpretable-by-design; see §B.1.

defined set of concepts [28, 29]. While effective, this approach does not generalize to unseen concepts and requires annotations that are costly to obtain for large datasets with many concepts. In *concept-specific supervised recovery*, a model is trained to predict whether a given concept is present in a given image. While this approach also requires annotated data, by leveraging vision-language embeddings such as CLIP [55], it can generalize to new concepts at inference time [9]. *Zero-shot concept recovery* methods rely solely on pre-trained vision-language models to predict concepts without requiring additional annotations [45], albeit at the cost of runtime efficiency. Finally, *sparse concept recovery* methods start with a predefined set of concepts and identify those present in an image by representing the image embedding as a sparse linear combination of the concept embeddings [4, 8]. This approach is particularly scalable because it focuses on selecting a small number concepts.

However, the aforementioned methods fail to capture hierarchical relationships among concepts, such as those between hypernyms and hyponyms. For instance, in Fig. 1, the concept hierarchy takes the form of a tree whose leaves correspond to the object classes. In this case, there is a unique path connecting each class to the root of the tree, and the concepts along that path can be viewed as an explanation for the class. For example, the explanation for `cat` in Fig. 1 is given by the path `animal` $\rightarrow$ `mammal` $\rightarrow$ `cat`. By neglecting this hierarchy, existing concept recovery methods may identify concepts that are inconsistent with it, leading to unreliable explanations and predictions. We address this limitation by drawing inspiration from prior work on the geometry of hierarchical concepts in large language models [53] and from hierarchical sparse coding [25, 27, 35]. This leads to a hierarchical sparse coding formulation for concept recovery that produces more reliable explanations.

Contributions. In this paper, we propose Hierarchical Concept Embedding & Pursuit (HCEP), a framework that leverages hierarchical relationships between concepts to improve concept recovery for interpretable image classification. We summarize HCEP below and highlight our contributions.

- *Hierarchical Concept Embedding:* In §3, we propose a geometric framework for embedding a given hierarchy of semantic concepts. More specifically, we identify the following *ideal geometric properties* for hierarchical concept embeddings: embeddings of descendant synsets[3] should be close to those of their parent synsets, while embeddings of sibling synsets should be well separated. We also incorporate geometric conditions inspired by prior work on the hierarchical geometry of large language models [53]. We analyze these properties theoretically and verify them empirically in vision-language models.
- *Hierarchical Concept Pursuit:* In §4, we leverage the geometric properties of hierarchical concept embeddings to propose a concept recovery procedure that proceeds in two steps. First, it constructs a *hierarchical dictionary* from pre-trained vision-language embeddings by taking differences between the embeddings of parent and child synsets. Second, it leverages *hierarchical sparse coding* to recover a rooted path in the hierarchy, thereby recovering concepts for interpretable classification.
- *Interpretable Image Classification:* In §5, we demonstrate through experiments on synthetic and real-world datasets that HCEP outperforms sparse concept recovery baselines in concept recovery while maintaining competitive classification accuracy. In few-shot settings, HCEP outperforms all interpretable baselines in both classification accuracy and concept recovery. We also show that for datasets without a predefined hierarchy, we can construct a meaningful hierarchy using taxonomy induction methods [21, 39, 72] and still achieve improved concept recovery using our framework.

[3] A synset is a categorical concept that groups all synonyms together (e.g., `cats`). A concept is a higher level abstraction that includes synsets and differences of synsets. One can view synsets as nodes in a hierarchy and synset differences as edges.

2. Preliminaries

2.1. Interpretable-by-design classification models

Interpretable-by-design classification models often proceed in two steps: (1) predicting human-interpretable concepts from the input and (2) using this concept-based representation for classification. Depending on the approach, interpretability may arise either from the use of linear classifiers (e.g., in concept bottleneck models (CBMs) [29] the importance of a concept depends on the magnitude of the classifier coefficients) or from the selection of a small set of most relevant concepts for classification (e.g., via information gain [7] or semantic sparse coding [4, 8]). As a result, the main design space in these models lies in reliable concept prediction. CBMs use supervision to learn concept predictors; however, they are not scalable because they require additional labeled data. Semantic sparse coding approaches instead use pre-trained embeddings to extract concepts from data, with the goal of representing an image embedding as a sparse linear combination of concept embeddings. In this work, we focus on extracting concepts from pre-trained image embeddings via a sparse coding objective.

2.2. Sparse coding for concept extraction

Sparse coding aims to represent data as a sparse linear combination of basis elements, typically chosen from an overcomplete dictionary [18]. Given an input signal $\mathbf{x} \in \mathbb{R}^d$ and a dictionary $\mathbf{D} \in \mathbb{R}^{d \times k}$, sparse coding seeks a sparse vector $\mathbf{z} \in \mathbb{R}^k$ such that $\mathbf{x} \approx \mathbf{D}\mathbf{z}$, where the sparsity of $\mathbf{z}$ encourages the model to use only a few dictionary elements to

reconstruct the input signal. To find a sparse representation $\mathbf{z}$, we can solve the following optimization problem:

$$\min_{\mathbf{z}} \|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2^2 + \lambda \|\mathbf{z}\|_0, \tag{1}$$

where $\|\cdot\|_0$ denotes the ℓ_0 semi-norm and λ is a regularization parameter that controls the trade-off between reconstruction accuracy and sparsity. The optimization problem in (1) can be solved using various algorithms, such as orthogonal matching pursuit (OMP) [54] or basis pursuit (ℓ_1 relaxation) [11]. Intuitively, standard OMP finds a sparse solution greedily: starting from the input $\boldsymbol{x}$ as the residual error, it repeatedly selects the dictionary atom most aligned with the current residual, adds that atom to the support, recomputes the coefficients of the selected atoms by least squares, and then updates the residual by subtracting its projection onto the span of the selected atoms (see Alg. 1).

In interpretable image classification, the signal $\mathbf{x}$ is typically the embedding of an image obtained from a pre-trained model, the dictionary $\mathbf{D}$ consists of text embeddings corresponding to human-interpretable concepts, and the magnitudes of the sparse coefficients in $\boldsymbol{z}$ indicate the importance of the corresponding concepts for representing the image [4, 8, 19].[4] Thus, OMP can be viewed as iteratively asking which concept (atom) best explains what is left unexplained by the current explanation (residual) [8].

However, sparse coding treats all columns of $\boldsymbol{D}$ equally, and thus it neglects the hierarchical structure of the concept set. For example, the synset `vehicle` can be further divided into `car`, `truck`, and `motorcycle`, each of which can be further divided into more specific synsets. These concepts, which *implicitly* describe the difference between a fine-grained synset and its parent synset, correspond to the edges of the synset hierarchy. To capture this hierarchical structure, we need to extend the sparse coding framework to incorporate hierarchical relationships among concepts. As motivation for this goal, we next briefly discuss related work on geometric structure in vector embeddings.

2.3. Geometric Structures of Meanings in Vector Embeddings

A notable example of geometric structure in vector embeddings is Word2Vec [46], where contrastive semantic relationships can be captured through vector arithmetic. More recent work has explored linear and compositional structures in a vector space [26, 52, 63, 64]. Hierarchical compositional structures have also been studied for learning part-based representations [31, 59]. On the other hand, Park et al. [53] connect hierarchical class-based semantics to a geometric condition: making a concept more specific should not interfere with the coarser concept. This intuition can be written as

$$(\mathbf{v}_{\text{child}} - \mathbf{v}_{\text{parent}})^\top \mathbf{v}_{\text{parent}} = 0. \tag{2}$$

where $\mathbf{v}_{\text{parent}}$ and $\mathbf{v}_{\text{child}}$ denote representations along one branch of a hierarchy. In the next section, we build on this insight to design a hierarchical concept embedding model.

[4] Although not the initial motivation, the validity of this approach is supported by the linear representation and superposition hypotheses [17, 52, 58].

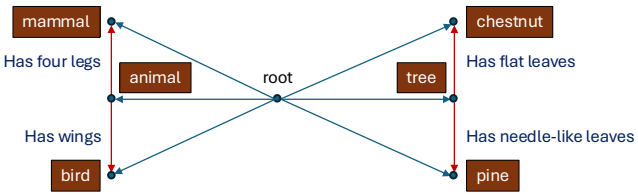

Figure 2. **Illustration of the hierarchical data model in $\mathbb{R}^2$.** The root node has two child nodes, `animal` and `tree`, each with its own children. There are two desirable conditions for concept identifiability: (1) children cluster around the parent while sibling nodes remain separated; and (2) the difference between a child and its parent (shown as red arrows) is orthogonal to the parent, and the differences between the children of a parent form a simplex (which is a line in $\mathbb{R}^2$). The difference vectors capture the characteristics that distinguish each child from its parent. See §3 for more details.

3. Hierarchical Concept Embedding

Given a hierarchy of semantic concepts, we seek vector representations whose geometry facilitates hierarchical concept recovery. In this section, we identify ideal geometric conditions that hierarchical concept embeddings should satisfy to enable such recovery, and later show that these conditions are empirically supported in vision-language models. The specific conditions we consider are the following:

- *Well-clustered synset embeddings (§3.1):* In the embedding space, sibling synsets should cluster sufficiently tightly around their parent, while remaining sufficiently well separated from one another to be easily distinguished. As a consequence, concepts with different ancestries are well separated and can be reliably recovered.
- *Hierarchical orthogonality (§3.2):* Given a parent node, the difference between a child's embedding and the parent's embedding should be orthogonal to the parent's embedding. This condition helps preserve semantic meaning across levels of the hierarchy in the embedding space.

Fig. 2 provides a two-dimensional illustration of the desired geometry: descendants cluster around their parents, while child-parent difference vectors capture refinements that are orthogonal to the parent's representation. These conditions drive the design of the concept recovery algorithm in §4.

3.1. Well-clustered synset embeddings

To ensure that each node in the hierarchy can be uniquely assigned to its parent, we impose geometric constraints on

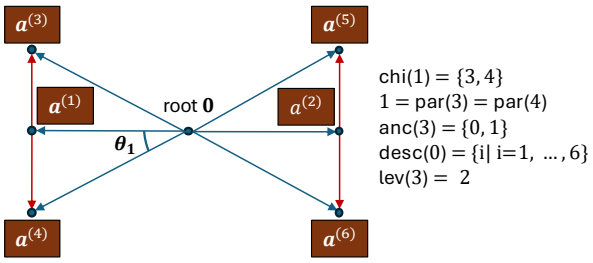

Figure 3. A hierarchy with $L = 2$ levels, branching ratio $b = 2$, and $N_L = 6$ nodes.

the embedding space. Specifically, consider a hierarchy of synsets, such as the one illustrated in Fig. 3. Let 0 denote the root node and $\mathcal{A} = \{1, \dots, N_L\}$ be the set of non-root nodes. Assume the hierarchy has L levels and that each internal node has b children. For node i, let $\mathrm{par}(i)$, $\mathrm{chi}(i)$, $\mathrm{anc}(i)$, and $\mathrm{desc}(i)$ denote its parent, children, ancestors, and descendants, respectively. The level of node i is the number of its ancestors, namely $\mathrm{lev}(i) = |\mathrm{anc}(i)|$. Our goal is to associate each node i with a vector representation $\boldsymbol{a}^{(i)} \in \mathbb{R}^d$. The following proposition formalizes conditions on $\{\boldsymbol{a}^{(i)}\}_{i=1}^{N_L}$ that guarantee well-separated subtrees.

Proposition 3.1 (Well-clustered hierarchy ensures unique parent assignment)**.** *Suppose the following geometric conditions hold for all nodes in the hierarchy:*

1. ***Subtree containment:*** *The vector representations of the descendants of node i, $\{\boldsymbol{a}^{(j)}\}_{j\in\mathrm{desc}(i)}$, are contained in a cone with axis $\boldsymbol{a}^{(i)}$ and half-angle $\theta_{\mathrm{lev}(i)}$, i.e.,*

$$\forall i \in \mathcal{A}, \quad \max_{j\in\mathrm{desc}(i)} \angle(\boldsymbol{a}^{(i)}, \boldsymbol{a}^{(j)}) \le \theta_{\mathrm{lev}(i)}. \tag{3}$$

2. ***Sibling-cone disjointness:*** *For any parent node i and any pair of distinct children $j, j' \in \mathrm{chi}(i)$, their corresponding subtree cones do not intersect:*

$$\angle(\boldsymbol{a}^{(j)}, \boldsymbol{a}^{(j')}) > \theta_{\mathrm{lev}(j)} + \theta_{\mathrm{lev}(j')}. \tag{4}$$

Then the subtrees rooted at any two sibling nodes are disjoint, and every node has a unique parent.

All proofs are provided in App. D. A sufficient way to satisfy both conditions in Prop. 3.1 is to impose a geometrically decreasing half-angle schedule, as presented next.

Proposition 3.2 (Geometrically decreasing half-angle schedule)**.** *If the half-angles satisfy*

$$\theta_{l+1} \le \min\{r, 1/b\}\, \theta_l \tag{5}$$

for some $r \in (0, 1/2)$, there exists a placement of node embeddings such that the conditions in Prop. 3.1 are satisfied.

3.2. Hierarchical Orthogonality

Inspired by the geometric conditions on concept embeddings in large language models [53], we further impose hierarchical orthogonality and simplex conditions on the concept embeddings.

- A parent embedding is orthogonal to the child-parent difference vector. That is, for a parent node i, any child node $j \in \mathrm{chi}(i)$ satisfies $(\boldsymbol{a}^{(j)} - \boldsymbol{a}^{(i)})^\top \boldsymbol{a}^{(i)} = 0$.
- The differences between the embeddings of two children of node i, $\{\boldsymbol{a}^{(j)} - \boldsymbol{a}^{(j')}\}_{j,j'\in\mathrm{chi}(i)}$, form a $(b-1)$-simplex.

As we shall see in the next section, these conditions will guide the construction of the dictionary we will use for hierarchical sparse coding. However, for them to hold, the embedding space must satisfy a minimum dimension requirement, as stated in the following proposition.

Proposition 3.3 (Depth–branch–dimension necessity)**.** *In a hierarchy with depth L and branching ratio b, suppose all nodes satisfy hierarchical orthogonality and all sibling nodes form a regular $(b-1)$-simplex as in (30) and (31). Then the embedding dimension must satisfy $d \ge L + b - 1$.*

Intuitively, at depth $l \in [L-1]$, hierarchical orthogonality imposes $l+1$ independent affine constraints, restricting children to a $(d-l-1)$-dimensional feasible subspace. Embedding a regular $(b-1)$-simplex within that subspace requires $(d-l-1) \ge (b-1)$. Evaluating this at the deepest internal level $l = L-1$ yields $d \ge L + b - 1$. This condition is satisfied in practice. For example, CLIP embeddings have $d = 768 \gg 38$ ($14 + 25 - 1$ for WordNet's depth and maximum branching ratio).

4. Hierarchical Concept Pursuit

Given the concept embedding model described in §3, we now turn to the problem of recovering the sparse representation of a signal generated from this model. To do so, we first construct a hierarchical dictionary that captures the hierarchical structure of the synsets and concepts (§4.1). We then propose a hierarchical sparse coding algorithm that leverages this structure to improve sparse recovery (§4.2).

4.1. Hierarchical Concept Dictionary Construction

First, we define the hierarchical dictionary $\boldsymbol{D}$ as

$$\boldsymbol{D} = \left[\, \boldsymbol{a}^{(j)} - \boldsymbol{a}^{(\mathrm{par}(j))} \,\right]_{j\in\mathcal{A}}. \tag{6}$$

Each column captures the difference between a synset and its parent. We define $\boldsymbol{a}^{(\mathrm{root})} = \boldsymbol{0}$, so the atoms for root children reduce to $\boldsymbol{a}^{(j)} - \boldsymbol{0} = \boldsymbol{a}^{(j)}$.

In the context of interpretable image classification, $\boldsymbol{a}^{(i)}$ represents the embedding of a synset within a concept hierarchy, such as WordNet [47], while the difference $\boldsymbol{a}^{(i)} - \boldsymbol{a}^{(\mathrm{par}(i))}$ represents the concept that distinguishes the synset

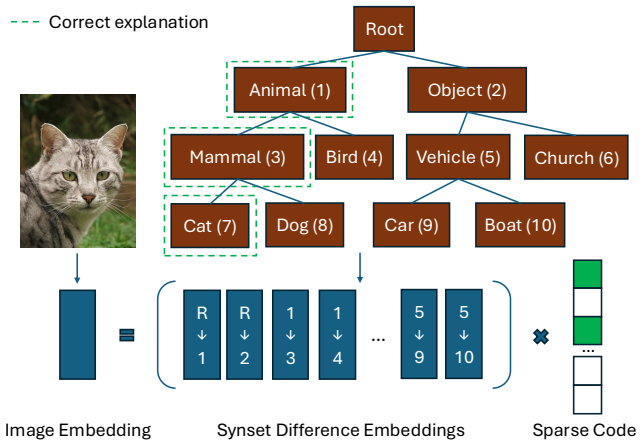

Figure 4. **Illustration of the hierarchical sparse decomposition in (8).** Given an image of a cat, the correct explanation follows the path from the root to the leaf node `Cat` (shown in green dashed lines). The image embedding is expressed as a sparse linear combination of the root child synset embedding (e.g., `Animal`) and synset difference embeddings along the path (e.g., `Mammal` − `Animal`, `Cat` − `Mammal`). The sparse code has non-zero entries only for atoms corresponding to nodes on the correct path.

from its parent. For example, let node i denote the synset `bear` and let node $j \in \mathrm{chi}(i)$ denote the synset `polar bear`. Then the difference $\boldsymbol{a}^{(j)} - \boldsymbol{a}^{(i)}$ can be interpreted as the refinement that distinguishes `polar bear` from `bear`, such as `white fur`. This construction of the dictionary in (6) avoids trivial solutions in the sparse formulation of concept recovery; for example, the sparsest explanation for an image of a `cat` would otherwise be simply the concept `cat`. Note that the difference embeddings have a grounded and interpretable meaning on their own, as they represent directions that differentiate child synsets from their parents. An illustration of this hierarchical sparse decomposition is shown in Fig. 4.

Under this hierarchical dictionary, a synset embedding can be written as a sparse linear combination of atoms along the root-to-node path:

$$\boldsymbol{a}^{(i)} = \sum_{j \in \mathrm{anc}(i) \cup \{i\}} \left(\boldsymbol{a}^{(j)} - \boldsymbol{a}^{(\mathrm{par}(j))}\right), \tag{7}$$

which follows by telescoping. Since $\boldsymbol{a}^{(\mathrm{root})} = \boldsymbol{0}$, the coefficients along the path are all equal to 1 in the noiseless case. If an image embedding $\boldsymbol{x}$ is generated from synset i with additive noise, i.e., $\boldsymbol{x} = \boldsymbol{a}^{(i)} + \epsilon$, then we expect a noisy expansion with coefficients $\alpha_j \in \mathbb{R}$:

$$\boldsymbol{x} = \sum_{j \in \mathrm{anc}(i) \cup \{i\}} \alpha_j \left(\boldsymbol{a}^{(j)} - \boldsymbol{a}^{(\mathrm{par}(j))}\right). \tag{8}$$

Recovering the synset i is therefore equivalent to recovering a sparse code whose support corresponds to the nodes on the root-to-i path, although the coefficients may deviate from 1.

Algorithm 1 Orthogonal Matching Pursuit (OMP)

Require: $\boldsymbol{x} \in \mathbb{R}^d$, dict $\boldsymbol{D}$, tol ϵ, max steps T

Init residual error $\boldsymbol{r}^{(0)} \leftarrow \boldsymbol{x}$, sparse code $\boldsymbol{z}^{(0)} \leftarrow \boldsymbol{0}$
for $t = 1, \ldots, T$ **do**
  if $\|\boldsymbol{r}^{(t-1)}\|_2 < \epsilon$ **then**
    break
  $i^\star \leftarrow \arg\max_i \left| \frac{\langle \boldsymbol{r}^{(t-1)}, \boldsymbol{d}^{(i)} \rangle}{\|\boldsymbol{r}^{(t-1)}\|_2 \, \|\boldsymbol{d}^{(i)}\|_2} \right|$
  $\mathcal{S}^{(t)} \leftarrow \mathrm{supp}(\boldsymbol{z}^{(t-1)}) \cup \{i^\star\}$ ▷ extend support
  $\boldsymbol{z}^{(t)} \leftarrow \underset{\boldsymbol{w}:\mathrm{supp}(\boldsymbol{w}) \subseteq \mathcal{S}^{(t)}}{\mathrm{argmin}} \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{w}\|_2^2$ ▷ update code
  $\boldsymbol{r}^{(t)} \leftarrow \boldsymbol{x} - \boldsymbol{D}\boldsymbol{z}^{(t)}$ ▷ update residual
Return $\boldsymbol{z}^{(t^\star)}$ where $t^\star \in \arg\min_t \|\boldsymbol{r}^{(t)}\|_2$

Algorithm 2 Hierarchical Beam OMP (HB-OMP)

Require: $\boldsymbol{x} \in \mathbb{R}^d$, dict $\boldsymbol{D}$, tol ϵ, max steps T, children map $\mathrm{chi}(\cdot)$, beam size B, residual error vector $\boldsymbol{r}_h$ for hypothesis h

Init with a null hypothesis set: $\mathcal{H}^{(0)} \leftarrow \left\{ \left(\boldsymbol{0}, \boldsymbol{x}, \mathrm{root}\right) \right\}$
▷ each hypothesis consists of (sparse code, residual, last node index)
for $t = 1, \ldots, T$ **do**
  if $\min_{h \in \mathcal{H}^{(t-1)}} \|\boldsymbol{r}_h\|_2 < \epsilon$ **then**
    break ▷ some hypothesis have a small residual
  $\mathcal{H}_{\mathrm{new}} \leftarrow \emptyset$
  for $h = (\boldsymbol{z}, \boldsymbol{r}, i_{last})$ in $\mathcal{H}^{(t-1)}$ **do**
    $\mathcal{I}_{\mathrm{active}} \leftarrow \mathrm{chi}(i_{last})$
    $c_i \leftarrow \left| \frac{\langle \boldsymbol{r}, \boldsymbol{d}^{(i)} \rangle}{\|\boldsymbol{r}\|_2 \, \|\boldsymbol{d}^{(i)}\|_2} \right|$ for all $i \in \mathcal{I}_{\mathrm{active}}$
    $\mathcal{C} \leftarrow$ top-B indices of c_i in $\mathcal{I}_{\mathrm{active}}$
    $\mathcal{H}_{\mathrm{new}} \leftarrow \mathcal{H}_{\mathrm{new}} \cup \textsc{ExtendHypo}(h, \mathcal{C}, \boldsymbol{x}, \boldsymbol{D})$
    ▷ Alg. 3
  $\mathcal{H}^{(t)} \leftarrow B$ lowest-$\|\boldsymbol{r}_h\|_2^2$ hypotheses in $\mathcal{H}_{\mathrm{new}}$
Return $\boldsymbol{z}_{h^\star}$ where $h^\star \in \arg\min_{h \in \mathcal{H}^{(t)}} \|\boldsymbol{r}_h\|_2$

4.2. Hierarchical Beam OMP (HB-OMP)

Given the hierarchical dictionary in (6), our goal is to recover the sparse code $\boldsymbol{z}$ whose support corresponds to the root-to-leaf path of the generating synset (7). As discussed in §2, standard OMP greedily selects the atom most aligned with the current residual and then recomputes the active coefficients. If OMP were applied to our hierarchical dictionary, it would treat all atoms as if they were at the same level of the hierarchy, selecting the most correlated atom at each step without respecting the hierarchy. Therefore, OMP may select atoms that violate the hierarchical structure (e.g., choosing both `animal` and `vehicle` at the same level).

In this work, we exploit the hierarchical structure with a coarse-to-fine search: at each step, we consider only chil-

Algorithm 3 Extend Hypothesis with Sparse Update

Require: hypothesis $h = (\boldsymbol{z}, \boldsymbol{r}, i_{last})$, candidates $\mathcal{C}$, signal $\boldsymbol{x}$, dict $\boldsymbol{D}$

$\mathcal{H}_{\text{new}} \leftarrow \emptyset$
if $\mathcal{C} = \emptyset$ **then** ▷ leaf reached because no children
  Add h to $\mathcal{H}_{\text{new}}$ ▷ keep hypothesis
  return $\mathcal{H}_{\text{new}}$
for each $i \in \mathcal{C}$ **do**
  $\mathcal{S}' \leftarrow \text{supp}(\boldsymbol{z}) \cup \{i\}$ ▷ extend support
  $\boldsymbol{z}' \leftarrow \underset{\boldsymbol{w}:\text{supp}(\boldsymbol{w})\subseteq\mathcal{S}'}{\text{argmin}} \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{w}\|_2^2$ ▷ update code
  $\boldsymbol{r}' \leftarrow \boldsymbol{x} - \boldsymbol{D}\boldsymbol{z}'$ ▷ update residual
  Add $(\boldsymbol{z}', \boldsymbol{r}', i)$ to $\mathcal{H}_{\text{new}}$
return $\mathcal{H}_{\text{new}}$

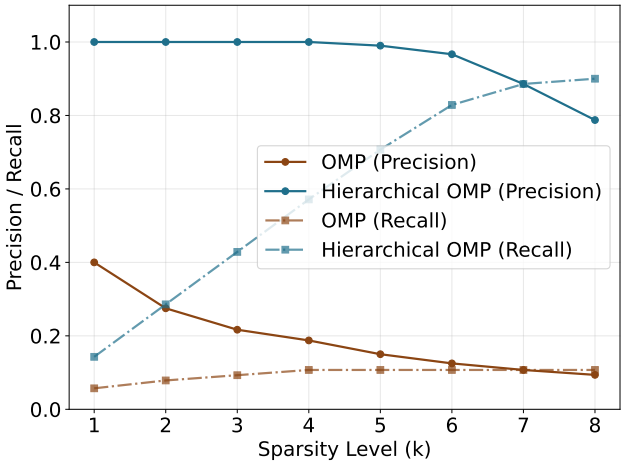

Figure 5. Hierarchical OMP has improved support recovery precision and recall compared to standard sparse coding methods on synthetic data.

dren of the current node, descending one level at a time. This yields Hierarchical Beam OMP (Alg. 2), a variant of Hierarchical OMP [27, 35] augmented with beam search. HB-OMP maintains a set of hierarchically valid support hypotheses, each a partial root-to-node path, and iteratively extends and prunes them according to reconstruction error. Relative to the standard OMP procedure introduced in §2.2 and detailed in Alg. 1, there are two key modifications:

1. Path-restricted extension: only children of the deepest explored node are eligible for selection in Alg. 2, restricting the search to hierarchically valid supports.
2. Beam search: the algorithm maintains the top-B hypotheses ranked by residual norm, mitigating error accumulation. An incorrect early decision (e.g., `animal` vs. `object`) does not propagate to all lower levels.

We next give an informal proposition explaining the source of this support-recovery advantage. See the formal statement and proof in App. D.5.

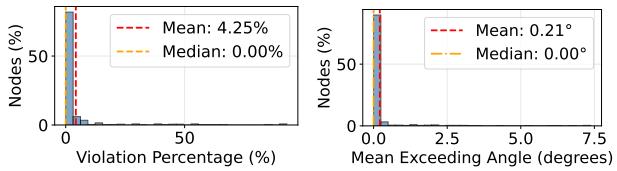

Figure 6. **CLIP image embeddings for ImageNet are tightly clustered.** (Left): For a fixed node i, we show the proportion of non-descendants of i whose embeddings intersect the cone of i. (Right): For the same node i, we show the mean angle violation, with non-violating angles counted as 0.

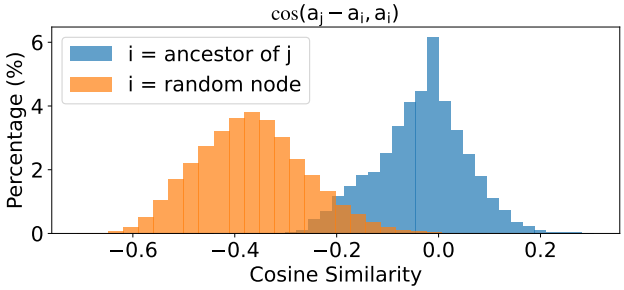

Figure 7. **Observed hierarchical orthogonality in CLIP image embeddings for ImageNet.** The cosine similarity between child-parent difference vectors and their parents is close to zero, while random pairings have significantly higher cosine similarity. This suggests that hierarchical orthogonality [53] holds even in pre-trained models.

Proposition 4.1 (Informal)**.** *Suppose that, at each iteration, HB-OMP extends a hypothesis whose support is a prefix of the true root-to-leaf path. Then its next selection is less likely to introduce an atom outside the true hierarchical support than OMP's selection.*

This result supports our choice of using beam search to explore multiple hypotheses, as it increases the likelihood of recovering the correct hierarchical support. Beam search also mitigates error accumulation in the top-down search: by maintaining multiple hypotheses at each level, an incorrect early decision (e.g., `animal` vs. `object`) does not necessarily propagate to all lower-level explanations.

5. Experiments

In this section, we show that HCEP improves hierarchical concept recovery on both synthetic (§5.1) and real-world datasets (§5.2). Across settings, the main empirical pattern is consistent: HB-OMP recovers the correct support more accurately than OMP, and these gains are especially pronounced in few-shot regimes where the hierarchy provides a strong inductive bias.

5.1. Synthetic Experiments

We first compare the performance of HB-OMP (Alg. 2) with OMP (Alg. 1) on synthetic data generated from the Hierarchical Concept Embeddings model in §3. We evaluate the

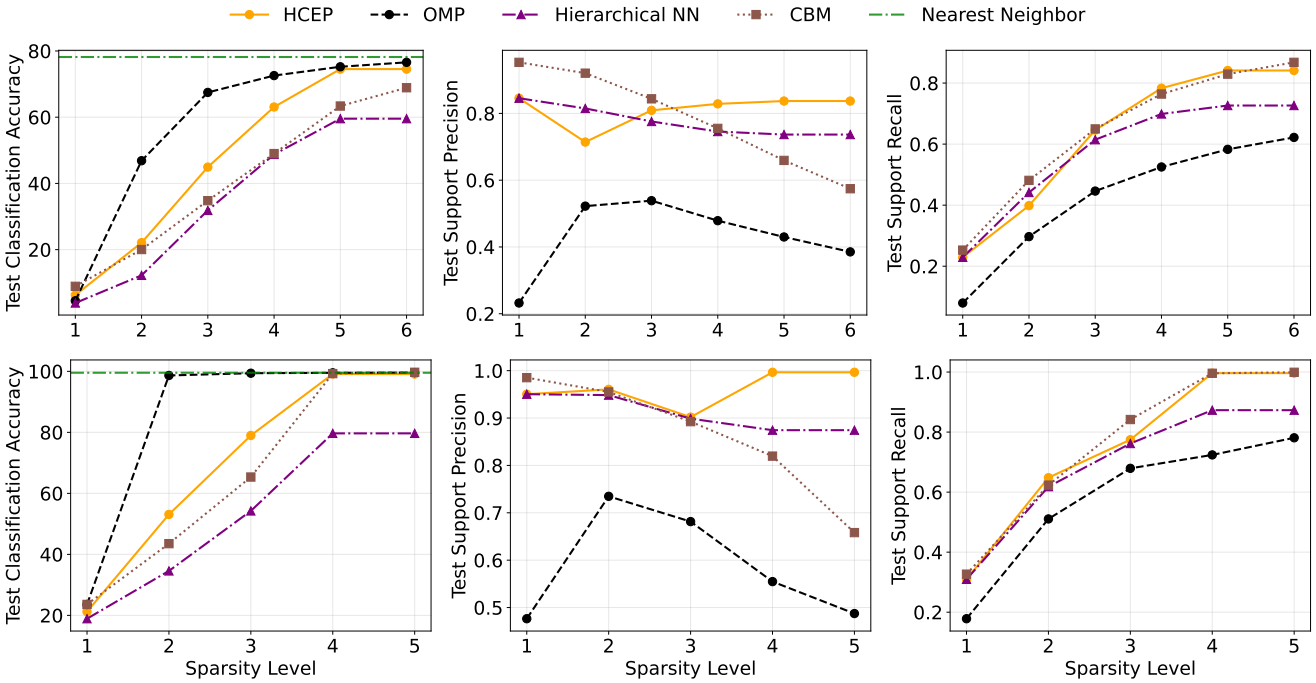

Figure 8. Interpretable image classification on CIFAR-100 (top) and ImageNette (bottom). HCEP achieves state-of-the-art support precision and recall while maintaining comparable classification accuracy at a sparsity level matching the depth of the hierarchy (5 for CIFAR-100, 4 for ImageNette).

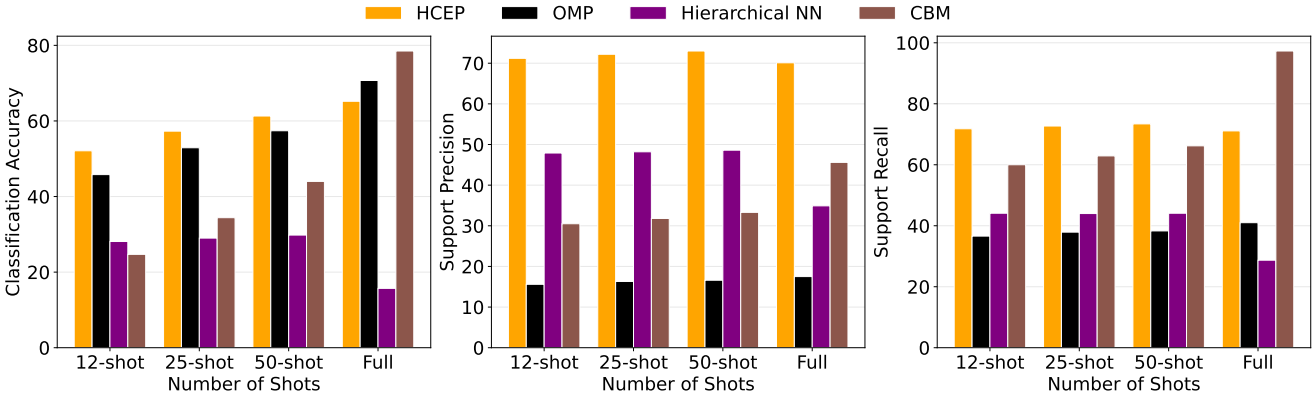

Figure 9. As we reduce the number of images per class in ImageNet, HCEP consistently improves test classification accuracy, support precision, and support recall over all baselines. Results are shown at sparsity level 14.

reconstruction error and the recovery of the ground-truth sparse support (i.e., the path from the root to the leaf node) under varying noise levels and hierarchy depths.

We choose a branching ratio of $b = 3$, a hierarchy depth of $L = 7$, and a dimension $d = 50$. Note that this dimension satisfies the depth-banch-dimension condition in Prop. 3.3 since $d = 50 \geq 7+3 = 10$. This gives us $2,187$ leaf synsets and $3,280$ atoms in the dictionary. We generate 5 samples per leaf for a total of $10,935$ samples. Further details on how the synthetic data is generated can be found in App. E, and the hyperparameters can be found in App. F.1.

As it can be observed in Fig. 5, HB-OMP consistently outperforms OMP in both precision and recall. This demonstrates the effectiveness of incorporating hierarchical structure into sparse coding for improved concept recovery.

5.2. Real-World Experiments

In this section, we evaluate HCEP on real-world image classification tasks. We first describe how we construct the hierarchy and dictionary, then present the experimental setting and results. More details are provided in App. F.3.

Datasets. We evaluate on (1) ImageNette [22], (2) ImageNet [14], and (3) CIFAR-100 [32]. For ImageNet-based datasets, we use the WordNet hierarchy, which has $L = 14$ levels with branching ratios up to $b = 25$, demonstrating that HCEP can handle complex, large-scale hierarchies. For

CIFAR-100, we use taxonomy induction methods [72] to construct a hierarchy over the classes.

Hierarchy and Dictionary Construction. We obtain the embedding of a class, which is a leaf node, as the average of the CLIP image embeddings for that class. We obtain the embedding of non-leaf synsets as the average of their children's embeddings. We construct the hierarchical dictionary as in (6) and keep it fixed throughout the experiments.

Baselines. (1) OMP [8, 54] with the full dictionary; (2) CBMs [29] which use supervised concept annotations to learn a concept predictor; (3) a Nearest Neighbor (NN) classifier that uses the class synset embeddings directly, which can be viewed as a black-box classifier; and (4) Hierarchical NN (HNN), which traverses the hierarchy using nearest neighbor search at each level and serves as a hierarchical interpretable-by-design baseline.

Evaluation Metrics. We evaluate the models based on (1) classification accuracy; (2) support precision and recall, which measure how well the recovered sparse support matches the ground-truth path in the hierarchy. Note that we do *not* assume that each leaf has a unique root-to-leaf path: in DAG-structured hierarchies such as WordNet, a synset may have multiple parents, yielding several valid paths. HCEP explores and selects one of these paths, and our evaluation scores the recovered support against the closest among all valid root-to-leaf paths.

Classification Procedure. For each image, we compute its CLIP embedding, and each method produces its own intermediate representation: (1) OMP and HB-OMP recover a sparse code; (2) HNN recovers a 0/1 sparse code by going down a rooted path in the hierarchy using nearest neighbor search at each level and record the nodes along the path in the sparse code; and (3) CBMs produce scores for all atoms at once by training a classifier on top of the CLIP image embeddings. The recovered intermediate representations are then fed to a linear classifier trained on the training set to predict the class labels. For NN, we directly use the class synset embeddings to classify images without any intermediate representation.

Results. See the results on ImageNet (Fig. 9) and CIFAR-100/ImageNette (Fig. 8). Across all three datasets, HB-OMP achieves higher support precision and recall than the other interpretable baselines, indicating more accurate recovery of the relevant hierarchical concepts. These gains are more pronounced in the few-shot ImageNet setting, where HB-OMP outperforms all interpretable baselines in both classification accuracy and support precision/recall.

This pattern is consistent with our hypothesis that sparse coding methods are especially helpful when concepts must be estimated from few labeled examples: OMP and HB-OMP only require estimating synset embeddings by averaging image embeddings, whereas CBMs must learn a concept classifier from labels. This few-shot scenario is common in real-world applications where labeled data is scarce but a pre-trained vision-language model can provide rich embeddings. Moreover, restricting the search to hierarchically valid paths further reduces spurious atom selections, so HB-OMP performs better than OMP in few-shot settings.

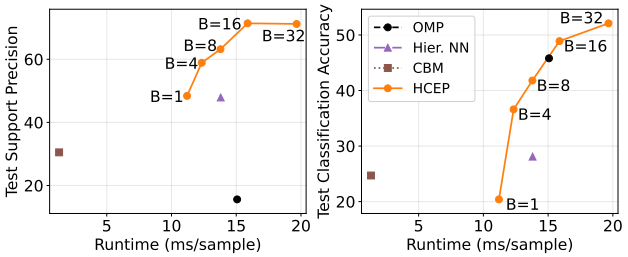

Figure 10. Runtime, support precision and classification accuracy on ImageNet. All methods use 12-shot and sparsity level 14.

We also verified that HCEP generalizes across vision-language models: replacing CLIP with SigLIP [73] on ImageNet yields similar improvements in interpretability metrics (see Fig. 12).

Well-Clustered Synsets and Hierarchical Orthogonality in Real-World Datasets. We empirically verify the geometric conditions described in §3 on real-world datasets. We focus on ImageNet [14] and use CLIP image embeddings as the representation space for the synsets and concepts. For the well-clustered synset condition (Prop. 3.1), Fig. 6 shows that most branches are tightly clustered and well separated from other branches. To test the validity of the hierarchical orthogonality condition in §3.2, we measure the cosine similarity between child-parent difference vectors and their parent embeddings. As shown in Fig. 7 for ImageNet (see Fig. 15 for CIFAR-100), the child-parent difference vectors are close to orthogonal to the parent vectors, whereas random pairings are not.

Runtime Analysis. Fig. 10 reports a runtime analysis using the same setting as Fig. 9. As HCEP's beam width increases from 1 to 32, support precision improves with only a modest runtime overhead, enabled by parallelizing the beam-search hypotheses on GPU.

6. Conclusion

We introduced HCEP, a geometric framework for hierarchical concept embeddings and a pursuit algorithm that respects the structure of synset hierarchies. We analyzed identifiability requirements through well-clustered cones, hierarchical orthogonality, and simplex structure, along with support-recovery guarantees for Hierarchical OMP. Empirically, the resulting codes deliver better support precision and recall than interpretable baselines across synthetic and real-world benchmarks. Our findings highlight the promise of structured sparse coding as a scalable and flexible framework for interpretable machine learning.

Acknowledgements

The authors thank Ryan Chan, Ryan Pilgrim, Uday Kiran Tadipatri, Buyun Liang, Kyle Poe, and anonymous reviewers for their helpful comments and suggestions. This work was funded by Simons Foundation grant 814201, NSF grant 2031985, and University of Pennsylvania startup funds.

References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *NeurIPS*, 2018. 1

[2] David Alvarez-Melis and Tommi S. Jaakkola. Towards robust interpretability with self-explaining neural networks. In *NeurIPS*, 2018. 1

[3] Fazl Barez, Tung-Yu Wu, Iván Arcuschin, Michael Lan, Vincent Wang, Noah Siegel, Nicolas Collignon, Clement Neo, Isabelle Lee, Alasdair Paren, Adel Bibi, Robert Trager, Damiano Fornasiere, John Yan, Yanai Elazar, and Yoshua Bengio. Chain-of-thought is not explainability. Oxford Martin AI Governance Initiative working paper, 2025. 12

[4] Usha Bhalla, Alex Oesterling, Suraj Srinivas, Flavio P Calmon, and Himabindu Lakkaraju. Interpreting CLIP with sparse linear concept embeddings. In *NeurIPS*, 2024. 2, 3, 12

[5] Blair Bilodeau, Natasha Jaques, Pang Wei Koh, and Been Kim. Impossibility theorems for feature attribution. *PNAS*, 2024. 1

[6] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nicholas L Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Chris Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html. 12

[7] Aditya Chattopadhyay, Kwan Ho Ryan Chan, Benjamin D Haeffele, Donald Geman, and René Vidal. Variational information pursuit for interpretable predictions. In *ICLR*, 2023. 2, 12

[8] Aditya Chattopadhyay, Ryan Pilgrim, and René Vidal. Information maximization perspective of orthogonal matching pursuit with applications to explainable AI. In *NeurIPS*, 2023. 2, 3, 8, 12

[9] Aditya Chattopadhyay, Kwan Ho Ryan Chan, and René Vidal. Bootstrapping variational information pursuit with large language and vision models for interpretable image classification. In *ICLR*, 2024. 2, 12

[10] Chaofan Chen, Oscar Li, Chaofan Tao, Alina Jade Barnett, Jonathan Su, and Cynthia Rudin. This looks like that: Deep learning for interpretable image recognition. In *NeurIPS*, 2019. 1

[11] Scott Shaobing Chen, David L Donoho, and Michael A Saunders. Atomic decomposition by basis pursuit. *SIAM Review*, 2001. 3, 12

[12] Valérie Costa, Thomas Fel, Ekdeep Singh Lubana, Bahareh Tolooshams, and Demba E. Ba. From flat to hierarchical: Extracting sparse representations with matching pursuit. In *NeurIPS*, 2025. 12

[13] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *arXiv preprint arXiv:2309.08600*, 2023. 12

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 7, 8

[15] Karan Desai, Maximilian Nickel, Tanmay Rajpurohit, Justin Johnson, and Ramakrishna Vedantam. Hyperbolic Image-Text Representations. In *ICML*, 2023. 13

[16] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. In *arXiv preprint arXiv:1702.08608*, 2017. 1

[17] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy Models of Superposition. In *arXiv preprint arXiv:2209.10652*, 2022. 3

[18] Simon Foucart and Holger Rauhut. *A Mathematical Introduction to Compressive Sensing*. Springer, 2013. 2

[19] Yossi Gandelsman, Alexei A. Efros, and Jacob Steinhardt. Interpreting CLIP's Image Representation via Text-Based Decomposition. In *arXiv preprint arXiv:2310.05916*, 2024. 3, 12

[20] Donald Geman and Bruno Jedynak. An active testing model for tracking roads in satellite images. *TPAMI*, 1996. 12

[21] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *COLING*, 1992. 2

[22] Jeremy Howard. Imagenette. GitHub repository, 2019. Accessed 2026-03-22. 7

[23] Aya Abdelsalam Ismail, Julius Adebayo, Hector Corrada Bravo, Stephen Ra, and Kyunghyun Cho. Concept bottleneck generative models. In *ICLR*, 2024. 12

[24] Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *ACL*, 2020. 12

[25] Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, and Francis Bach. Proximal methods for hierarchical sparse coding. *JMLR*, 2011. 2, 12, 13

[26] Yibo Jiang, Goutham Rajendran, Pradeep Ravikumar, Bryon Aragam, and Victor Veitch. On the origins of linear representations in large language models. In *arXiv preprint arXiv:2403.03867*, 2024. 3

[27] Philippe Jost, Pierre Vandergheynst, and Pascal Frossard. Tree-Based Pursuit: Algorithm and Properties. *IEEE Transactions on Signal Processing*, 2006. 2, 6, 12

[28] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *ICML*, 2018. 2, 12

[29] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept Bottleneck Models. In *ICML*, 2020. 1, 2, 8, 12

[30] Stefan Kolek, Aditya Chattopadhyay, Kwan Ho Ryan Chan, Hector Andrade-Loarca, Gitta Kutyniok, and René Vidal. Learning interpretable queries for explainable image classification with information pursuit. In *ICCV*, 2025. 12

[31] Adam Kortylewski, Aleksander Wieczorek, Mario Wieser, Clemens Blumer, Sonali Parbhoo, Andreas Morel-Forster, Volker Roth, and Thomas Vetter. Greedy structure learning of hierarchical compositional models. In *CVPR*, 2019. 3

[32] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. 7

[33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 1

[34] Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, and Shree K. Nayar. Attribute and simile classifiers for face verification. In *ICCV*, 2009. 12

[35] Chinh La and Minh N. Do. Tree-Based Orthogonal Matching Pursuit Algorithm for Signal Reconstruction. In *ICIP*, Atlanta, GA, 2006. 2, 6, 12

[36] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009. 12

[37] Curtis P. Langlotz. Radlex: A new method for indexing online educational materials. *RadioGraphics*, 2006. 13

[38] Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. Measuring Faithfulness in Chain-of-Thought Reasoning. In *arXiv preprint arXiv:2307.13702*, 2023. 12

[39] Matthew Le, Stephen Roller, Laetitia Papaxanthos, Douwe Kiela, and Maximilian Nickel. Inferring Concept Hierarchies from Text Corpora via Hyperbolic Embeddings. In *ACL*, Florence, Italy, 2019. 2

[40] Zachary C. Lipton. The mythos of model interpretability. *Communications of the ACM*, 2018. 1

[41] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *arXiv preprint arXiv:1711.05101*, 2017. 17

[42] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *NeurIPS*, 2017. 1

[43] Julien Mairal, Guillermo Sapiro, and Michael Elad. Learning multiscale sparse representations for image and video restoration. *Multiscale Modeling & Simulation*, 2008. 12

[44] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *JMLR*, 2010. 12

[45] Sachit Menon and Carl Vondrick. Visual Classification via Description from Large Language Models. In *ICLR*, 2023. 2

[46] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, 2013. 3

[47] George A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 1995. 4, 17

[48] Andrew Ng. Sparse autoencoder. Stanford CS294A lecture notes, 2011. Stanford University. 12

[49] Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *NeurIPS*, 2017. 13

[50] Matthew Lyle Olson, Musashi Hinck, Neale Ratzlaff, Changbai Li, Phillip Howard, Vasudev Lal, and Shao-Yen Tseng. Analyzing Hierarchical Structure in Vision Models with Sparse Autoencoders. In *arXiv preprint arXiv:2505.15970*, 2025. 12

[51] OpenAI. GPT-5 System Card. Technical report, OpenAI, 2025. Accessed 2026-03-22. 17

[52] Kiho Park, Yo Joong Choe, and Victor Veitch. The Linear Representation Hypothesis and the Geometry of Large Language Models. In *arXiv preprint arXiv:2311.03658*, 2024. 3

[53] Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. The Geometry of Categorical and Hierarchical Concepts in Large Language Models. In *arXiv preprint arXiv:2406.01506*, 2025. 2, 3, 4, 6

[54] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Asilomar Conference on Signals, Systems and Computers*, 1993. 3, 8, 12

[55] 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 *arXiv preprint arXiv:2103.00020*, 2021. 1, 2, 17

[56] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *KDD*, 2016. 1

[57] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 2019. 1

[58] Baturay Saglam, Paul Kassianik, Blaine Nelson, Sajana Weerawardhena, Yaron Singer, and Amin Karbasi. Large language models encode semantics and alignment in linearly separable representations. In *arXiv preprint arXiv:2507.09709*, 2025. 3

[59] Antonio Sclocchi, Alessandro Favero, Noam Itzhak Levi, and Matthieu Wyart. Probing the latent hierarchical structure of data via diffusion models. In *arXiv preprint arXiv:2410.13770*, 2024. 3

[60] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017. 1

[61] Chung-En Sun, Tuomas Oikarinen, Berk Ustun, and Tsui-Wei Weng. Concept bottleneck large language models. In *ICLR*, 2025. 12

[62] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *ICML*, 2017. 1

[63] Matthew Trager, Pramuditha Perera, Luca Zancato, Alessandro Achille, Parminder Bhatia, and Stefano Soatto. Linear spaces of meanings: Compositional structures in vision-language models. In *ICCV*, 2023. 3

[64] Matthew Trager, Alessandro Achille, Pramuditha Perera, Luca Zancato, and Stefano Soatto. Compositional structures in neural embedding and interaction decompositions. In *arXiv preprint arXiv:2407.08934*, 2024. 3

[65] Joel A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 2004. 15

[66] Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *NeurIPS*, 2023. 12

[67] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *CVPR*, 2018. 13

[68] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*, 2022. 1, 12

[69] John Wright, Allen Y. Yang, Arvind Ganesh, S. Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *TPAMI*, 2009. 13

[70] Abdurrahim Yilmaz, Sirin Pekcan Yasar, Gulsum Gencoglan, and Burak Temelkuran. Derm12345: A large, multisource dermatoscopic skin lesion dataset with 40 subclasses. *Scientific Data*, 2024. 13

[71] Vladimir Zaigrajew, Hubert Baniecki, and Przemyslaw Biecek. Interpreting clip with hierarchical sparse autoencoders. In *ICML*, 2025. 12

[72] Qingkai Zeng, Yuyang Bai, Zhaoxuan Tan, Shangbin Feng, Zhenwen Liang, Zhihan Zhang, and Meng Jiang. Chain-of-layer: Iteratively prompting large language models for taxonomy induction from limited examples. In *CIKM*, 2024. 2, 8, 13, 18

[73] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023. 8, 17

Contents

A. Extended Related Work

Interpretable-by-design models. Interpretable-by-design models aim to provide explanations for their predictions by using human-interpretable concepts as intermediate representations. Early works explored attribute-based classification for face verification [34] and learning to detect unseen object classes through attribute transfer [36]. Subsequent works include Concept Activation Vectors [28], which use linear classifiers to identify directions in the embedding space corresponding to specific concepts. Concept Bottleneck Models [29] extend this idea by training a model to predict concepts before predicting the final output. In an adjacent line of work, Information Pursuit [20] is used as a criterion to choose the most relevant concepts [7, 9, 30]. More recent works have explored leveraging pre-trained embeddings and sparse coding for identifying specific concept directions [4, 8, 19]. Our work builds upon these foundations by introducing a concept embedding framework that captures the hierarchical relationships among synsets in interpretable image classification.

Sparse Recovery. Sparse recovery aims to recover a sparse signal from a set of observations, often using techniques such as Orthogonal Matching Pursuit (OMP) [54] and Basis Pursuit [11]. Sparse coding has been widely used in image processing [43, 44], signal processing, and machine learning. Although there have been works on hierarchical sparse coding [25, 27, 35], they do not consider the hierarchical structure of concepts in the context of interpretable models or deep representation learning. More recently, sparse autoencoders (SAEs) [6, 13, 48] have been used in vision-language models [12, 50, 71] to interpret the structure of concepts beyond sparsity. Unlike SAE-based approaches, our work focuses on structured sparse coding with an explicitly hierarchical dictionary derived from semantic synset relationships.

B. Preliminaries

B.1. Interpretable-by-design vs. Chain-of-Thought

Chain-of-Thought (CoT) prompting [68] is a technique that enables large language models to generate step-by-step reasoning traces before producing a final answer, often improving performance on complex reasoning tasks. However, using CoT as an interpretability method does not guarantee faithfulness [5] because the final prediction is conditioned on both the input and the generated chain of thoughts. Several recent audits further question the faithfulness of CoT explanations [3, 38, 66]. Therefore, the development of CoT does not make interpretable-by-design models obsolete. In fact, there is a trend toward making foundation models (e.g., LLMs or diffusion models) interpretable-by-design by enforcing an interpretable concept bottleneck in the latent space [23, 61].

[5] An explanation is *faithful* if it accurately reflects the true computation process to the final prediction [24].

B.2. Canonical regular simplex

Define

$$\tilde{\boldsymbol{s}}_j \;=\; \boldsymbol{e}_j \;-\; \frac{1}{b}\,\mathbf{1}, \qquad j = 1, \ldots, b, \tag{9}$$

where $\{\boldsymbol{e}_j\}_{j=1}^{b}$ are the standard basis vectors of $\mathbb{R}^b$ and $\mathbf{1} \in \mathbb{R}^b$ is the all-ones vector. These centred vertices satisfy $\sum_{j=1}^{b} \tilde{\boldsymbol{s}}_j \;=\; \mathbf{0}$ and $\tilde{\boldsymbol{s}}_j^{\top} \tilde{\boldsymbol{s}}_k \;=\; \begin{cases} 1, & j = k, \\ -\dfrac{1}{b-1}, & j \neq k, \end{cases}$ i.e. they form a regular $(b-1)$-simplex of unit edge length in $\mathbb{R}^{b-1}$.

C. Limitations

While our framework demonstrates clear advantages in concept recovery, it also has several limitations:

Embedding Constraints. Our theoretical analysis (Prop. 3.3) establishes that embedding a hierarchy with leaf depth L and branching ratio b requires ambient dimension $d \geq L + b - 1$. For deep hierarchies (large L) or highly branching structures (large b), this constraint becomes restrictive. Real-world embeddings from models such as CLIP typically have fixed dimensions (e.g., $d = 768$), which limits the depth and complexity of hierarchies that can be faithfully represented. Moreover, as hierarchies deepen, the half-angles of the cones containing each subtree (Prop. 3.2) must decrease geometrically. As mentioned in §3.1, this limitation may necessitate exploring alternative geometries, such as hyperbolic spaces [15, 49], for more faithful hierarchical representations. At the same time, by using pretrained embeddings, HCEP provides accurate and interpretable classification without the additional compute required to finetune large models, and pretrained models are available and improving across multiple domains, making HCEP easy to extend. That said, hierarchy-aware finetuning could further improve the geometric conditions in §3 and thus boost HCEP's performance; we leave this direction to future work.

Hierarchy Quality Dependence. The performance of Hierarchical OMP critically depends on the quality of the predefined hierarchy. For ImageNet-based datasets, we leverage the well-curated WordNet hierarchy, which provides semantically meaningful relationships. However, for CIFAR-100, we rely on taxonomy induction methods [72], which may produce hierarchies with inconsistencies or unclear relationships. That said, high-quality hierarchies already exist in many domains beyond common objects, e.g., RadLex [37] for radiology, DERM12345 [70] for skin lesions, and iNaturalist [67] for species classification. Moreover, the quality of LLM-based taxonomy induction is steadily improving, as demonstrated by our CIFAR-100 experiments (§5.2). When the hierarchy is noisy, HCEP may produce incorrect explanation paths. A promising direction for future work is to interpolate between hierarchical and non-hierarchical solutions via ℓ_1 minimization with a hierarchy regularizer [25], thereby allowing the method to degrade gracefully when hierarchy quality is uncertain.

Computational Complexity. Hierarchical OMP with beam search (Alg. 2) has complexity $O(TBb|\mathcal{D}_{\text{active}}|)$, where T is the number of iterations, B is the beam width, b is the branching ratio, and $|\mathcal{D}_{\text{active}}|$ is the size of the active dictionary at each level. In contrast, OMP has complexity $O(T|\mathcal{D}|)$, where $|\mathcal{D}|$ is the size of the entire dictionary. For large branching ratios or deep hierarchies, this may become computationally expensive. While we demonstrate better concept recovery accuracy over standard OMP, the computational cost remains a practical consideration for deployment at scale. In practice, the beam search hypotheses can be parallelized on GPU, which significantly reduces wall-clock time overhead (see Fig. 10).

Outlier Handling. HCEP assumes that the input belongs to a leaf class in the provided hierarchy. For outliers within a known class (e.g., a cat with three legs), classical results on the robustness of sparse coding to corruption [69] suggest that the outlier will still be closest to the correct synset path, while a larger sparse reconstruction error can flag such cases. If a novel class is entirely absent from the hierarchy, taxonomy induction can be used to append the new class before applying HCEP.

D. Proofs

D.1. Proof of Prop. 3.1

Statement. If subtree containment (Eq. (3)) and sibling-cone disjointness (Eq. (4)) hold, then the subtrees rooted at sibling nodes do not overlap.

Proof. Suppose j and j' are distinct children of a parent node i. We show that their subtrees are disjoint.

Let $k \in \operatorname{desc}(j)$. By subtree containment in Eq. (3),

$$\angle(\boldsymbol{a}^{(j)}, \boldsymbol{a}^{(k)}) \leq \theta_{\text{lev}(j)}. \tag{10}$$

Assume for contradiction that $k \in \operatorname{desc}(j')$ as well. Applying Eq. (3) to j' gives

$$\angle(\boldsymbol{a}^{(j')}, \boldsymbol{a}^{(k)}) \leq \theta_{\text{lev}(j')}. \tag{11}$$

Now consider the spherical triangle formed by the unit vectors $\boldsymbol{a}^{(j)}/\|\boldsymbol{a}^{(j)}\|$, $\boldsymbol{a}^{(k)}/\|\boldsymbol{a}^{(k)}\|$, and $\boldsymbol{a}^{(j')}/\|\boldsymbol{a}^{(j')}\|$. The triangle inequality yields

$$\begin{aligned} \angle(\boldsymbol{a}^{(j)}, \boldsymbol{a}^{(j')}) &\leq \angle(\boldsymbol{a}^{(j)}, \boldsymbol{a}^{(k)}) + \angle(\boldsymbol{a}^{(k)}, \boldsymbol{a}^{(j')}) \\ &\leq \theta_{\text{lev}(j)} + \theta_{\text{lev}(j')}, \end{aligned} \tag{12}$$

which contradicts sibling-cone disjointness in Eq. (4). Therefore k cannot belong to both $\operatorname{desc}(j)$ and $\operatorname{desc}(j')$.

Since j and j' were arbitrary siblings, the subtrees rooted at sibling nodes are disjoint. □

D.2. Proof of Prop. 3.2

Statement. If the half-angles satisfy $\theta_{l+1} \leq \min\{r, 1/b\}\,\theta_l$ with $r \in (0, 1/2)$, then there exists a placement of the nodes such that *subtree containment* (Eq. (3)) and *sibling-cone disjointness* (Eq. (4)) hold.

Proof. We verify the two requirements separately.

Step 1: Subtree containment. A sufficient condition for Eq. (3) is that the cumulative half-angles of all lower levels do not exceed the half-angle budget at level l, namely

$$\sum_{k=l+1}^{L} \theta_k \leq \theta_l, \quad \forall i \in \{1, \dots, N_L\},\ l = \mathrm{lev}(i). \tag{13}$$

Assume first that the half-angles satisfy the geometric decrease

$$\theta_{l+1} \leq r\,\theta_l. \tag{14}$$

Then, for any level l,

$$\sum_{k=l+1}^{L} \theta_k \leq \sum_{k=1}^{L-l} \theta_l\, r^k \text{(by Eq. (14))} \tag{15}$$

$$= \theta_l \sum_{k=1}^{L-l} r^k \tag{16}$$

$$= \theta_l\, r \sum_{k=0}^{L-l-1} r^k \tag{17}$$

$$\leq \theta_l\, r\, \frac{1 - r^{L-l-1}}{1 - r} \text{(sum of a geometric series).} \tag{18}$$

Taking $L \to \infty$ yields

$$\theta_l\, r \frac{1 - r^{L-l-1}}{1 - r} \to \theta_l\, \frac{r}{1 - r} \leq \theta_l, \qquad r < \frac{1}{2},$$

so Eq. (13) holds. This proves subtree containment.

Step 2: Sibling-cone disjointness. Next, we show that Eq. (4) follows from the simpler sufficient condition

$$\theta_{l+1} \leq \frac{1}{b}\,\theta_l. \tag{19}$$

Consider any 2D plane containing the parent cone axis. In that plane, a cone of half-angle α appears as a planar angle of magnitude 2α. Therefore, one can place b child cone axes inside the parent cone without overlap whenever b child angles of size $2\theta_{l+1}$ fit inside the parent angle $2\theta_l$, i.e., whenever $b\theta_{l+1} \leq \theta_l$. This is exactly Eq. (19).

Since the assumption of the proposition is $\theta_{l+1} \leq \min\{r, 1/b\}\theta_l$, both Step 1 and Step 2 hold simultaneously. Hence there exists a placement satisfying subtree containment and sibling-cone disjointness. □

D.3. Proof of Prop. 3.3

Statement. Under the hierarchical orthogonality constraints in Eq. (30) and the regular-simplex difference condition in Eq. (31) at every internal node up to depth $L-1$ of a hierarchy whose leaves are at depth L in ambient space $\mathbb{R}^d$, it is necessary that the ambient dimension satisfies the depth–dimension condition $d \geq L + b - 1$.

Proof. Fix a level $l \geq 0$ and consider the ancestor path $\boldsymbol{A}_l = \{\boldsymbol{a}^{(\pi_0)}, \dots, \boldsymbol{a}^{(\pi_l)}\}$. The hierarchical orthogonality constraints in Eq. (30) define the affine feasible set for child candidates:

$$\mathcal{V}_l = \{\boldsymbol{x} \in \mathbb{R}^d : \boldsymbol{A}_l^\top \boldsymbol{x} = \boldsymbol{h}_l\},$$

which is Eq. (34). When the ancestor vectors are linearly independent, as is generically the case because each level introduces a new non-collinear direction, we have

$$\dim \mathcal{V}_l = d - (l + 1).$$

Now apply the regular-simplex condition in Eq. (31). It requires placing b child points whose differences relative to a feasible origin in $\mathcal{V}_l$ form a regular $(b-1)$-simplex. Since such a simplex has affine hull of dimension $b-1$, it can be embedded in $\mathcal{V}_l$ only if

$$\dim \mathcal{V}_l \ \geq\ b - 1.$$

Combining the two displays yields, for every level l, $d - (l+1) \geq b - 1 \iff d \geq l + b$. Requiring this inequality to hold at the deepest internal level $l = L - 1$ gives $d \geq (L-1) + b = L + b - 1$, as claimed. □

D.4. Intermediate results for Prop. 4.1

Lemma D.1 (Column normalization equivalence)**.** *Let $\boldsymbol{D} = [\boldsymbol{d}_1, \dots, \boldsymbol{d}_k] \in \mathbb{R}^{d \times k}$ with arbitrary nonzero column norms ($\|\boldsymbol{d}_j\|_2 > 0$ for all $j \in [k]$), and define the diagonal matrix $\boldsymbol{W} := \mathrm{diag}(\|\boldsymbol{d}_1\|_2, \dots, \|\boldsymbol{d}_k\|_2)$ and the column-normalized dictionary $\widehat{\boldsymbol{D}} := \boldsymbol{D}\,\boldsymbol{W}^{-1}$. For any s-sparse $\boldsymbol{z} \in \mathbb{R}^k$ with support S, set $\widehat{\boldsymbol{z}} := \boldsymbol{W}\boldsymbol{z}$. Then $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{z} = \widehat{\boldsymbol{D}}\,\widehat{\boldsymbol{z}}$ and $\mathrm{supp}(\widehat{\boldsymbol{z}}) = S$. Moreover, OMP run on $\boldsymbol{D}$ with the selection rule*

$$j^\star \in \arg\max_j \frac{\left|\langle \boldsymbol{r}, \boldsymbol{d}_j \rangle\right|}{\|\boldsymbol{d}_j\|_2} = \arg\max_j \frac{\left|\langle \boldsymbol{r}, \boldsymbol{d}_j \rangle\right|}{\|\boldsymbol{d}_j\|_2\, \|\boldsymbol{r}\|_2} \tag{20}$$

is identical (same index picked at every iteration) to OMP run on $\widehat{\boldsymbol{D}}$ with the usual (unnormalized) correlation rule. Equivalently, this selects the atom with the highest absolute cosine similarity to the residual.

Proof. The representation identity is immediate:

$$\boldsymbol{x} = \boldsymbol{D}\boldsymbol{z} = \boldsymbol{D}\,\boldsymbol{W}^{-1}\boldsymbol{W}\boldsymbol{z} = \widehat{\boldsymbol{D}}\,\widehat{\boldsymbol{z}}, \tag{21}$$

and clearly $\mathrm{supp}(\widehat{\boldsymbol{z}}) = S$ because $\boldsymbol{W}$ is diagonal with positive entries.

For the selection rule, note that $\widehat{\boldsymbol{d}}_j = \boldsymbol{d}_j / \|\boldsymbol{d}_j\|_2$, so

$$\langle \boldsymbol{r}, \widehat{\boldsymbol{d}}_j \rangle = \frac{\langle \boldsymbol{r}, \boldsymbol{d}_j \rangle}{\|\boldsymbol{d}_j\|_2}. \tag{22}$$

Thus maximizing correlation with $\widehat{\boldsymbol{d}}_j$ is exactly the same as maximizing normalized correlation, equivalently absolute cosine similarity, with $\boldsymbol{d}_j$.

Finally, diagonal rescaling of the columns in $\boldsymbol{D}_S$ does not change $\mathrm{span}(\boldsymbol{D}_S)$, so the orthogonal projector onto this span is invariant under the rescaling $\boldsymbol{D}_S \mapsto \widehat{\boldsymbol{D}}_S$. Therefore, once the same index is selected, both versions of OMP produce the same least-squares fit and hence the same residual at every step. By induction over the iterations, the selected indices are identical throughout the run. □

Definition D.2 (Exact Recovery Coefficient (ERC) on normalized dictionary)**.** [6] For a support S with $\widehat{\boldsymbol{D}}_S$ full column rank, define

$$\mathrm{ERC}(\widehat{\boldsymbol{D}}; S) := \max_{j \in S^c} \big\| \widehat{\boldsymbol{D}}_S^\dagger \widehat{\boldsymbol{d}}_j \big\|_1, \tag{23}$$

$$\mathrm{ERC}(\widehat{\boldsymbol{D}}; S \,|\, T) := \max_{j \in T \setminus S} \big\| \widehat{\boldsymbol{D}}_S^\dagger \widehat{\boldsymbol{d}}_j \big\|_1, \tag{24}$$

for any $T \supseteq S$, where $\|\cdot\|_1$ denotes the vector ℓ_1 norm. Here $S^c := [k] \setminus S$, so $T \setminus S = T \cap S^c \subseteq S^c$; thus $\mathrm{ERC}(\widehat{\boldsymbol{D}}; S \,|\, T)$ is the same maximum as $\mathrm{ERC}(\widehat{\boldsymbol{D}}; S)$, but over the smaller index set $T \setminus S$.

Lemma D.3 (Monotone ERC improvement under subtree restriction)**.** *Let $\boldsymbol{D} \in \mathbb{R}^{d \times k}$ have arbitrary nonzero column norms and let $\boldsymbol{z}$ be s-sparse with support S. Let $T_0 \supset T_1 \supset \cdots \supset T_L$ be a nested sequence with $T_0 = [k]$ and $S \subseteq T_\ell$ for all $\ell = 0, \ldots, L$. Assume $\widehat{\boldsymbol{D}}_S$ has full column rank. Then the ERC decreases monotonically along the restriction:*

$$\mathrm{ERC}(\widehat{\boldsymbol{D}}; S \,|\, T_L) \le \mathrm{ERC}(\widehat{\boldsymbol{D}}; S \,|\, T_{L-1}) \tag{25}$$

$$\le \cdots \le \mathrm{ERC}(\widehat{\boldsymbol{D}}; S \,|\, T_0) \tag{26}$$

$$:= \mathrm{ERC}(\widehat{\boldsymbol{D}}; S). \tag{27}$$

Proof. By definition, $\mathrm{ERC}(\widehat{\boldsymbol{D}}; S \,|\, T) = \max_{j \in T \setminus S} \|\widehat{\boldsymbol{D}}_S^\dagger \widehat{\boldsymbol{d}}_j\|_1$. Since $\widehat{\boldsymbol{D}}_S^\dagger$ is fixed, shrinking T only restricts the index set over which this maximum is taken, so the value cannot increase. □

Lemma D.4 (ERC threshold implies restricted OMP success)**.** *Under the assumptions of Lemma D.3, if $\mathrm{ERC}(\widehat{\boldsymbol{D}}; S \,|\, T_L) < 1$, then OMP run on $\boldsymbol{D}$ with the normalized selection rule*

$$j^* \in \arg\max_j \frac{|\langle \boldsymbol{r}, \boldsymbol{d}_j \rangle|}{\|\boldsymbol{d}_j\|_2} = \arg\max_j \frac{|\langle \boldsymbol{r}, \boldsymbol{d}_j \rangle|}{\|\boldsymbol{d}_j\|_2 \, \|\boldsymbol{r}\|_2}, \tag{28}$$

restricted to T_L, recovers S in s steps. Equivalently, OMP on $\widehat{\boldsymbol{D}}_{T_L}$ with the standard rule succeeds in s iterations.

Proof. Lemma D.1 shows that the normalized-selection rule on $\boldsymbol{D}$ matches standard OMP on $\widehat{\boldsymbol{D}}$. The classical noiseless ERC theorem of Tropp [65] applied to the restricted dictionary $\widehat{\boldsymbol{D}}_{T_L}$ then yields exact support recovery in s iterations whenever $\max_{j \in T_L \setminus S} \|\widehat{\boldsymbol{D}}_S^\dagger \widehat{\boldsymbol{d}}_j\|_1 < 1$, equivalently whenever $\mathrm{ERC}(\widehat{\boldsymbol{D}}; S \,|\, T_L) < 1$. □

D.5. Proof of Prop. 4.1

Statement. There exist instances with $\mathrm{ERC}(\widehat{\boldsymbol{D}}; S) \ge 1$ yet $\mathrm{ERC}(\widehat{\boldsymbol{D}}; S \,|\, T_L) < 1$ for some nested $T_0 \supset T_1 \supset \cdots \supset T_L$ satisfying the right-subtree assumption $S \subseteq T_\ell$. Consequently, hierarchical OMP yields a strictly larger ERC-certified success region than global OMP on the full dictionary.

Proof. Choose any instance for which the maximizer of $\mathrm{ERC}(\widehat{\boldsymbol{D}}; S)$ lies outside T_L. Removing that maximizer from the admissible index set strictly decreases the maximum, so

$$\mathrm{ERC}(\widehat{\boldsymbol{D}}; S \,|\, T_L) < \mathrm{ERC}(\widehat{\boldsymbol{D}}; S).$$

Hence it is possible to have

$$\mathrm{ERC}(\widehat{\boldsymbol{D}}; S) \ge 1 \qquad \text{but} \qquad \mathrm{ERC}(\widehat{\boldsymbol{D}}; S \,|\, T_L) < 1.$$

Whenever this occurs, Lemma D.4 certifies exact recovery for Hierarchical OMP on the restricted set T_L, while the standard ERC guarantee for global OMP on the full dictionary does not apply. □

E. Step-by-step construction of a Hierarchical Concept Embedding

Assume we are at depth $l > 0$ of the hierarchy. The path from the root to the *current parent* π_l (with embedding $\boldsymbol{a}^{(\pi_l)} \in \mathbb{R}^d$) consists of the $l+1$ ancestor vectors

$$\boldsymbol{A}_l = \{\boldsymbol{a}^{(\pi_0)}, \boldsymbol{a}^{(\pi_1)}, \ldots, \boldsymbol{a}^{(\pi_l)}\}, \qquad \pi_0 < \pi_1 < \cdots < \pi_l. \tag{29}$$

We seek embeddings $\{\boldsymbol{a}^{(j)}\}_{j=1}^b \subset \mathbb{R}^d$ for its b children that satisfy the following conditions:

(i) **Hierarchical Orthogonality**:

$$(\boldsymbol{a}^{(j)} - \boldsymbol{a}^{(\pi_k)})^\top \boldsymbol{a}^{(\pi_k)} = 0, \quad k = 0, \ldots, l, \quad j = 1, \ldots, b. \tag{30}$$

[6] We use Exact Recovery Coefficient (ERC) for the quantity whose threshold at 1 is the classical exact recovery condition in Tropp [65].

By induction, the current parent node π_l already satisfies these orthogonality constraints with respect to its ancestors.

(ii) **Regular $(b-1)$-simplex structure**:

$$(\boldsymbol{a}^{(j)}-\boldsymbol{g}_l)^\top(\boldsymbol{a}^{(k)}-\boldsymbol{g}_l)=\begin{cases}\lambda_l^2, & j=k,\\ -\dfrac{\lambda_l^2}{b-1}, & j\neq k,\end{cases} \tag{31}$$

where $\boldsymbol{g}_l$ is any point satisfying all $l+1$ equations in Eq. (30), and λ_l scales the simplex so that $\angle(\boldsymbol{a}^{(j)},\boldsymbol{a}^{(\pi_l)})=\theta_l$.

(iii) **Cone condition w.r.t. the current parent**:

$$\begin{aligned}&\angle(\boldsymbol{a}^{(j)},\boldsymbol{a}^{(\pi_l)})=\theta_l\\ \iff &\|\boldsymbol{a}^{(j)}-\boldsymbol{a}^{(\pi_l)}\|=\|\boldsymbol{a}^{(\pi_l)}\|\tan\theta_l,\\ &j=1,\dots,b.\end{aligned} \tag{32}$$

This equivalence holds because Eq. (30) implies $\langle \boldsymbol{a}^{(j)}-\boldsymbol{a}^{(\pi_l)},\boldsymbol{a}^{(\pi_l)}\rangle=0$.

E.1. Feasible Subspace Induced by Hierarchical Orthogonality

Let

$$\boldsymbol{A}_l=\begin{bmatrix}\boldsymbol{a}^{(\pi_0)} & \boldsymbol{a}^{(\pi_1)} & \dots & \boldsymbol{a}^{(\pi_l)}\end{bmatrix}\in\mathbb{R}^{d\times(l+1)}. \tag{33}$$

The $l+1$ hyperplanes in (30) intersect in the affine subspace

$$\mathcal{V}_l=\{\boldsymbol{x}\in\mathbb{R}^d:\boldsymbol{A}_l^\top\boldsymbol{x}=\boldsymbol{h}_l\}, \tag{34}$$

where $\boldsymbol{h}_l=[\|\boldsymbol{a}^{(\pi_0)}\|^2,\dots,\|\boldsymbol{a}^{(\pi_l)}\|^2]^\top$. If the ancestor columns of $\boldsymbol{A}_l$ are linearly independent[7], then

$$\dim\mathcal{V}_l=d-(l+1). \tag{35}$$

To be able to embed a regular $(b-1)$-simplex for all depths we therefore require the *depth-dimension condition*

$$d\geq L+b. \tag{36}$$

Equation (36) quantifies the depth–dimension trade-off: one ambient degree of freedom is lost per additional ancestor constraint, while $(b-1)$ directions are always needed to accommodate the regular simplex of conditionally independent children.

We now give a constructive procedure for the children of node π_l at level l.

1. **Find one feasible origin.** Solve the linear system $\boldsymbol{A}_l^\top\boldsymbol{x}=\boldsymbol{h}_l$ to obtain any particular solution $\boldsymbol{g}_l\in\mathcal{V}_l$. If we also impose the cone half-angle condition, the feasible set can be further restricted to the intersection of $\mathcal{V}_l$ with the cone centered at the parent embedding $\boldsymbol{a}^{(\pi_l)}$ and half-angle θ_l from §3.

   A convenient choice, which preserves the full half-angle budget, is to take the current parent embedding itself as the feasible origin, i.e., $\boldsymbol{g}_l=\boldsymbol{a}^{(\pi_l)}$. By construction, every node is orthogonal to all of its ancestors. Hence, for each $k\in\{0,\dots,l\}$,

$$(\boldsymbol{a}^{(\pi_l)}-\boldsymbol{a}^{(\pi_k)})^\top\boldsymbol{a}^{(\pi_k)}=0 \tag{37}$$

$$\implies \boldsymbol{a}^{(\pi_l)\top}\boldsymbol{a}^{(\pi_k)}=\|\boldsymbol{a}^{(\pi_k)}\|^2. \tag{38}$$

2. **Basis for the difference linear space.** Compute an orthonormal basis

$$\boldsymbol{U}_l\in\mathbb{R}^{d\times(d-l-1)}, \tag{39}$$

$$\boldsymbol{A}_l^\top\boldsymbol{U}_l=\boldsymbol{0}, \tag{40}$$

$$\boldsymbol{U}_l^\top\boldsymbol{U}_l=I_{d-l-1}, \tag{41}$$

   e.g., by taking the $d-(l+1)$ left singular vectors of $\boldsymbol{A}_l$ associated with the smallest singular values, denoted $\boldsymbol{U}_{l+2:d}$.

3. **Canonical regular simplex in $\mathbb{R}^{b-1}$.** Use the centred construction $\tilde{\boldsymbol{d}}_i=\boldsymbol{e}_i-\frac{1}{b}\boldsymbol{1},\ i=1,\dots,b$ (cf. Eq. (9)).

4. **Scale $\tilde{\boldsymbol{d}}_j$ to satisfy the cone condition.**

$$\lambda_l=\|\boldsymbol{a}^{(\pi_l)}\|\tan\theta_l, \tag{42}$$

$$\boldsymbol{d}_j=\lambda_l\,\tilde{\boldsymbol{d}}_j. \tag{43}$$

5. **Embed and translate.**

$$\boldsymbol{a}^{(j)}=\boldsymbol{a}^{(\pi_l)}+\boldsymbol{U}_{l+2:d}\,\boldsymbol{d}_j,\qquad j=1,\dots,b. \tag{44}$$

   Choosing the scale factor

$$\lambda_l=\|\boldsymbol{a}^{(\pi_l)}\|\tan\theta_l, \tag{45}$$

forces $\|\boldsymbol{a}^{(j)}-\boldsymbol{a}^{(\pi_l)}\|=\|\boldsymbol{a}^{(\pi_l)}\|\tan\theta_l$ for all j, and therefore $\angle\big(\boldsymbol{a}^{(j)},\boldsymbol{a}^{(\pi_l)}\big)=\theta_l$, which is precisely the requirement in Eq. (32).

F. Additional Experimental Results

F.1. Additional Synthetic Experiment Details

Branching ratio $b=3$, hierarchy depth $L=7$, dimension $d=50$. Initial cone half angle = 85 degrees. Initial vector norm = 0.8. Geometric reduction factor = 0.4. Total leaf nodes = 2187. Total nodes = number of atoms = 3280. Gaussian noise for each leaf for data generation $\sigma^2=10^{-5}$. Generate 5 samples per leaf for a total of 10,935 samples.

[7] This is typical because every level adds a new non-collinear vector.

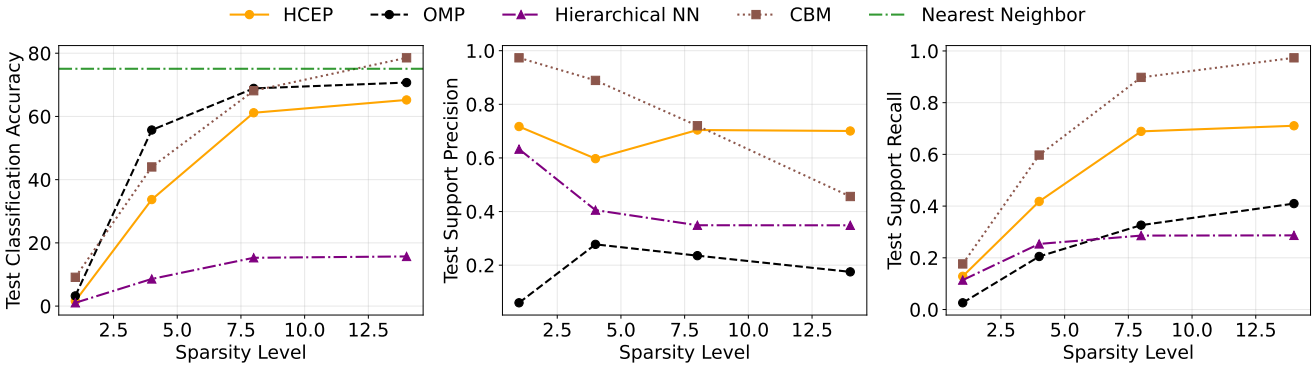

Figure 11. On ImageNet, HCEP achieves competitive accuracy while having higher concept precision/recall than sparse concept prediction baselines (OMP, Hierarchical NN).

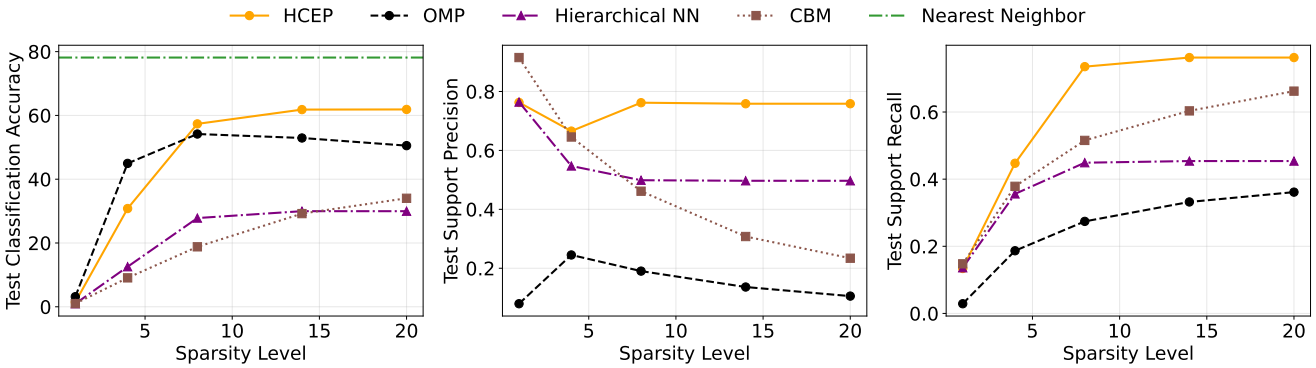

Figure 12. Interpretable image classification on ImageNet (12-shot) using SigLIP [73] embeddings. HCEP exhibits similar improvements in support precision and recall over baselines as with CLIP (cf. Fig. 11), demonstrating generality across vision-language models.

F.2. Text Interpretation of Synset Differences

Optionally, to qualitatively evaluate alternative textual meanings of the synset differences, which form the atoms in our hierarchical dictionary, we use CLIP text embeddings and GPT-5 [51]. First, for each parent-child pair, we generate a text description of the difference between the parent and child synsets using GPT-5. Then, we pool the text embeddings of these descriptions to form a set of candidate concept embeddings. Next, for each synset difference vector, we find its top-k neighbors among the candidate concept embeddings. Finally, we use GPT-5 to generate a summary of the top-k neighbor descriptions, which helps interpret the synset difference. Table 1 shows example interpretations for several parent-child pairs in WordNet [47], the hierarchy underlying ImageNet.

F.3. Additional Real-Data Experiment Details

Model Architecture and Training Details. We use CLIP-ViT-L/14 as the backbone. To train the linear classifier, we use the AdamW optimizer [41] with weight decay 10^{-4} and learning rate 10^{-1}. To train the CBM, we use the Adam optimizer with learning rate 10^{-1} for 500 epochs. We provide the detailed hyperparameters in Table 2.

Table 1. Example text interpretations of child–parent synset differences produced via CLIP embeddings and GPT summarization.

Parent→Child Pair	Text Interpretation
`bear` → `polar bear`	thick matte white fur blending with snow.
`container` → `basket`	open-top woven or perforated sides with handles.
`structure` → `lumbermill`	vertical log-sawing machines and plank conveyors.
`citrus` → `orange`	round, bright orange, pebbled rind.

Synset Difference Interpretations. We use CLIP text embeddings [55] and GPT-5 [51] to generate textual interpretations of the synset differences. We provide the top-10 concepts for each parent-child pair in Table 3. We also include the GPT-5 prompt in Table 5.

Ablation Study on Beam Size. We perform an ablation

study on the beam size for Hierarchical OMP. We vary the beam size from 1 to 8 and evaluate concept recovery accuracy on ImageNette. We report the results in Fig. 14.

Taxonomy Generation Prompt. We use the taxonomy generation prompt from Zeng et al. [72] in Table 4 to generate the taxonomy for CIFAR-100.

Table 2. Key hyperparameters used in experiments for each dataset.

Hyperparameter	**ImageNette**	**CIFAR-100**	**ImageNet**
Batch size	4096	4096	16384
Classification training epochs	500	500	1000
HCEP beam size	8	16	32

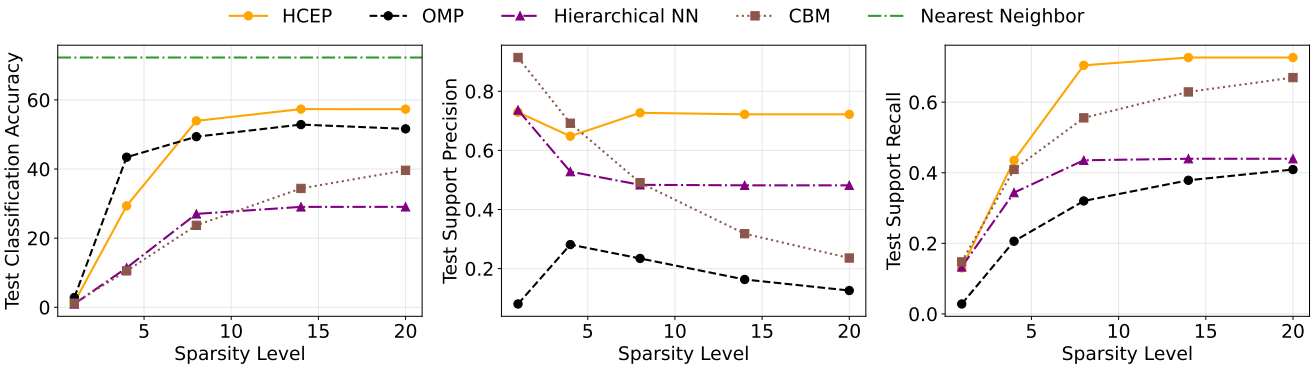

Figure 13. When we restrict the ImageNet training set to 25 images per class, HCEP outperforms all interpretable baselines.

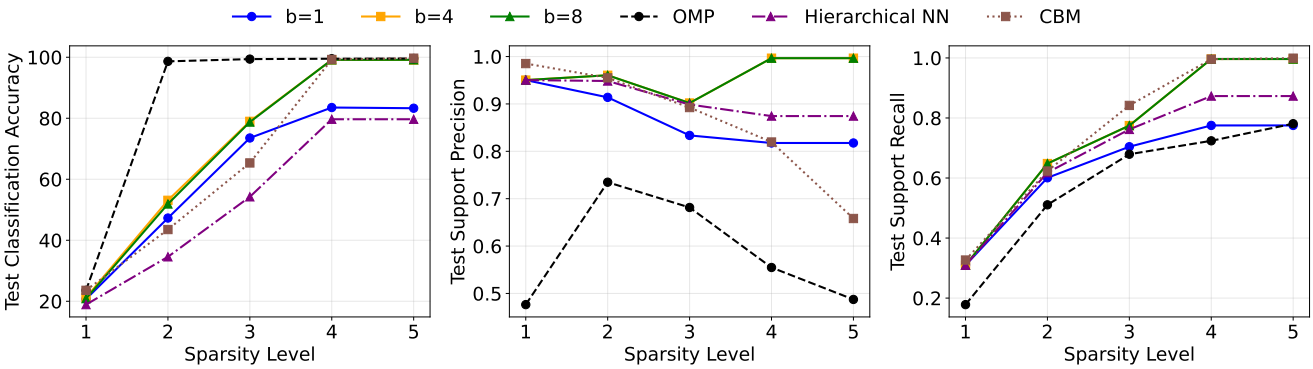

Figure 14. We vary the beam size over $\{1, 4, 8\}$ and evaluate on ImageNette.

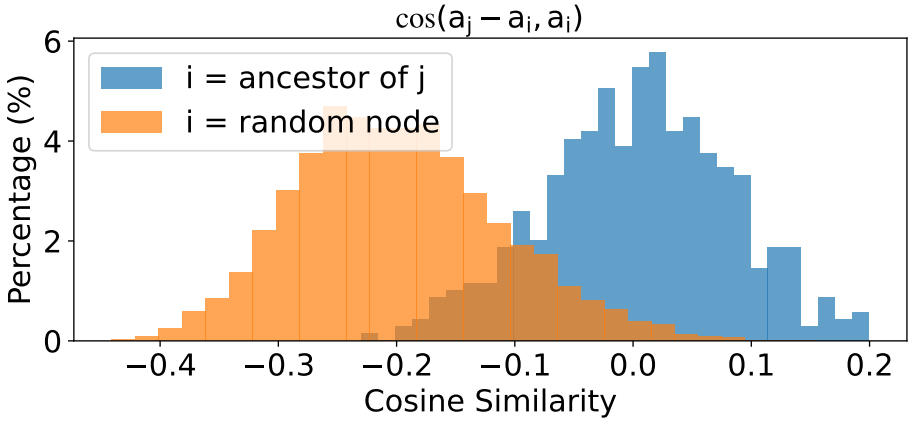

Figure 15. Observed hierarchical orthogonality on CIFAR-100. The cosine similarity between child-parent difference vectors and their parents is close to zero, while random non-parent pairs have significantly lower cosine similarity.

Table 3. Example text interpretations of child–parent synset differences produced via CLIP embeddings and GPT summarization, with the top-10 contributing concepts.

Parent→Child Pair	Top-10 Concepts	Text Interpretation
`bear` → `polar bear`	1. Matte white texture 2. Thick white winter fur coat 3. White plumage in winter season 4. White fur with cream patches 5. Long, silky white coat 6. White fur blending with surroundings 7. Pure white fluffy coat 8. White coat with lemon markings 9. White cue ball 10. Long, corded white coat	thick matte white fur blending with snow.
`container` → `basket`	1. Wicker baskets filled with baguettes 2. Rectangular shopping baskets 3. Stacked woven baskets 4. Rear storage basket 5. Rectangular open-top basket 6. Woven rattan backrest 7. Plastic shopping baskets 8. Perforated cutlery basket in lower rack 9. Woven rattan seating surfaces 10. Rectangular basket frame	open-top woven or perforated sides with handles.
`structure` → `lumbermill`	1. Vertical log slicing machines 2. Massive log cutting machines 3. Metallic sawmill machinery 4. Heavy-duty sawmill frames 5. Conveyor belts with wood pieces 6. Wooden board sorting systems 7. Exposed wooden axles 8. Wooden log feeding chutes 9. Stacks of cut wooden planks 10. Narrow wooden steering wheel	vertical log-sawing machines and plank conveyors.

Table 4. LLM prompt used for taxonomy generation from root and leaf concepts.

Taxonomy Generation Prompt

Given root concept <root> and leaf concepts <leaves>, generate a detailed hierarchical taxonomy that organizes these leaves under the root. Create multiple levels of intermediate category hierarchies to build a rich, fine-grained classification structure. Use as many hierarchical levels as needed to create meaningful semantic groupings and subgroupings.
The format is: 1. Parent Concept 1.1 Child Concept 1.1.1 Grandchild Concept.
CRITICAL: Every single leaf concept from the list must appear exactly as given in the taxonomy as the deepest level nodes. You may and should add multiple levels of intermediate concepts but do not add new leaf concepts. Before finishing, verify that each leaf concept from <leaves> appears in your taxonomy. Aim for depth and semantic richness in the hierarchy.

Table 5. LLM prompt used to generate child-vs-parent residual phrases.

Residual Concept Generation Prompt

TASK: Generate a concise phrase (3-10 words) that describes what distinguishes "<child>" from its parent category "<parent>". This is for a hierarchical sparse model. A residual represents the visual difference between a child and parent category. Most correlated visual concepts (from CLIP embeddings): <concepts_string> REQUIREMENTS:
1. Generate ONE short phrase (3-10 words) that captures the key distinguishing features
2. Base your phrase on the correlated concepts provided above
3. Focus on the most salient visual features
4. Be specific and concrete, not vague or generic
5. Use natural language that a human would use to describe the difference
6. IMPORTANT: Do NOT use the synset names ("<parent>" or "<child>") in your phrase
7. IMPORTANT: Describe only the visual features, not the category name
EXAMPLES OF GOOD PHRASES: - "tawny coat with distinctive facial markings" (for a specific dog breed) - "long curved neck and pink coloration" (for flamingo vs bird) - "striped pattern and elongated body" (for a specific fish) - "metallic surface with cylindrical shape" (for a lighter)