Formal Concept Lattices are Good Semantic Scaffolds for Concept-Based Learning

Anonymous Authors[1]

Abstract

Learning semantics is essential for deep learning models to be interpretable and better aligned with human reasoning. Concept-based models approach this by representing classes through meaningful semantic abstractions, but typically treat all concepts as a flat, unstructured set learned at a single neural network layer. This overlooks a fundamental property of human semantic understanding: concepts being organized hierarchically, from general to specific. While deep networks do learn a hierarchy of visual features, this structure is rarely aligned with explicit semantic hierarchies. Drawing on Formal Concept Analysis, we demonstrate that formal concept lattices provide principled semantic scaffolds to guide neural network learning. These lattices naturally identify where in the network concepts should be learned based on their level of generality. This allows the model to develop staged, semantically grounded representations throughout its depth. Empirical results on real-world datasets show that our models produce more interpretable embeddings, support more effective interventions, and learn concept representations that are both meaningful and hierarchically structured.

1. Introduction

For many years now, deep neural networks (DNNs) have been known to learn hierarchical representations. In computer vision, the early layers of these networks capture generic features like texture, while the later layers encode class-specific information (Zeiler & Fergus, 2014; Olah et al., 2018). However, the exact nature of these representations remains opaque and semantically less interpretable. Recent work has focused on making models inherently interpretable (Chen et al., 2019; Sarkar et al., 2022) so as to have better insight into what these models are learning. In particular, concept-based models or CBMs (Koh et al., 2020; Oikarinen et al., 2023; Liu et al., 2025) have emerged as a promising direction that quantify how learned concepts contribute to predictions. However, existing concept-based models typically learn all concepts at a single layer, overlooking the inherent hierarchical structure in neural network representations across multiple layers.

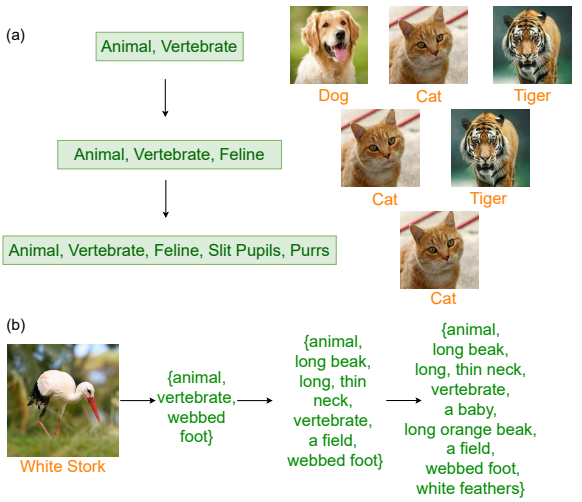


Figure 1. Classes (orange) and attributes (green) (a) An illustrative example of how attributes shared by more classes are more general; those shared by fewer are more specific, naturally forming a subset-superset hierarchy; (b) Attributes learned by a FoCA CBM at different layers for the class *White Stork* in ImageNet100.

Human cognition, on the other hand, organizes knowledge through semantic hierarchies and reasons over different levels of learned concepts (Theves et al., 2021) (see Fig 1a). Existing CBMs do not leverage such structure; while a few limited recent efforts have explored hierarchical concept sets (Panousis et al., 2024; Sun et al., 2024), architecturally, all concepts are still learned at the same stage in the network - immediately before classification. In contrast, in this work, we examine *how concept learning can capture hierarchical structure across network depth*. To this end, we explicitly guide the network to learn general human-understandable concepts in early layers and specific ones in deeper layers and see that this helps the model learn more semantically

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

grounded representations while additionally allowing interpretability at different granularities.

In classification tasks where each class is defined by a set of attributes, a natural semantic hierarchy emerges from the pattern of attribute sharing. Attributes shared by many classes are general, while those shared by a few are specific. For instance, a *cat* and *tiger* are an *animal, vertebrate* and are *feline*; while a *dog*, *cat* and *tiger* are an *animal* and are *vertebrate*. Here, the latter group $\{$*animal, vertebrate*$\}$ is more general as it spans more classes. Conversely, more specific attributes help better class discriminability in data samples. This subset-superset structure creates a concept hierarchy where generality corresponds to the number of classes sharing an attribute set, as shown in Fig 1.

To leverage such semantic structure while learning DNN models, we draw on Formal Concept Analysis (FCA) (Ganter & Wille, 2024) to construct a concept lattice from class-attribute associations. The lattice identifies natural supervision points in the network by aligning with class density patterns across network depth. Supervisory signals are then extracted from the lattice to learn sets of attributes and classes at these supervision points. These sets define *hierarchical semantic layers*, each comprising an attribute layer and a classifier layer, that effectively overlay a semantic scaffold onto the network's visual feature hierarchy. (Fig 1b illustrates the progressively refined attribute sets for the class *White Stork*). Each attribute layer corresponds to (and hence predicts) a group of classes, with the group's specificity determined by the granularity of its attributes. This layered structure enables progressive refinement of class predictions, while improving model transparency. One could view our formulation as generalizing extant CBM paradigms by exploiting a dataset's concept taxonomy. When the taxonomy is flat, it defaults to the canonical single-layer concept representation characteristic of traditional CBMs.

Our Contributions:

- We generalize the notion of concept-based interpretability in DNNs to semantic hierarchies using concept lattices, thus providing a means to leverage semantic scaffolds to guide learning across layers and enabling a deeper notion of interpretability in such models.
- We theoretically analyze our approach to study why such semantic ordering matters for concept-based models.
- Through comprehensive experiments on benchmark datasets, we show that this approach to learning not only performs on accuracy, but also yields semantically more meaningful embeddings, which we show using a clustering analysis.
- As part of a range of ablation studies and analysis, we show that our framework provides a mechanism to conduct multi-level concept interventions, going beyond the standard single-layer interventions in existing CBMs.

2. Related Work

Concept-Based Models. Building inherently interpretable models using concepts is an actively growing area of research, initially introduced as the idea of learning classes through a concept layer in the network (Koh et al., 2020). Follow-up works improve various aspects of these models, such as addressing concept leakage (Marconato et al., 2022), including uncertainty quantification (Kim et al., 2023) and improving robustness (Sinha et al., 2022). Other efforts include increasing model capacity using additional unsupervised concepts (Sawada & Nakamura, 2022) and building concept bases for such models (Yuksekgonul et al., 2023). More recent efforts have attempted the use of LLMs and VLMs for concept guidance and annotations (Oikarinen et al., 2023; Yang et al., 2023; Srivastava et al., 2024) and have studied concept relations (Vandenhirtz et al., 2024b; Raman et al., 2024). All these efforts focus on learning concepts at the last layer with *no semantic scaffolding across layers*. This is an aspect we focus on in this work.

Hierarchical Learning. Hierarchical learning has been explored more generally from a few perspectives. One line of work learns hierarchical embeddings like order embeddings (Vendrov et al., 2015), hyperbolic entailment cones (Ganea et al., 2018) and Poincaré embeddings (Nickel & Kiela, 2017). These methods, however, typically impose geometries *across samples* to align with pre-existing structure (often hierarchical) in the label space. Our approach, on the other hand, induces a concept hierarchy across features of *single sample*. Another line of work uses a hierarchy to constrain the predictions of the model (Giunchiglia & Lukasiewicz, 2020; 2022; Li et al., 2023). Some CBM-based works use hierarchical concept sets, although they are limited to two-level ones (Sun et al., 2024; Panousis et al., 2024) and are limited to the pre-classification step. In contrast, we focus on deriving supervisory signals from a structured formal concept lattice with an arbitrary number of levels (26 levels on one of the datasets we use).

Formal Concept Analysis (FCA). This is a mathematical theory of data analysis where a set of objects and attributes are used to derive a a structured hierarchy of formal concepts. There have been a few sparse efforts to use this theory in deep learning settings: to encode closure operators in a neural network (Rudolph, 2007), to introduce an embedding technique (Durrschnabel et al., 2019) for problems with formal context-like structures like bipartite graphs (Peng et al., 2024), and to obtain order-based representations using binary vectors (Gyurek et al., 2024). To the best of our knowledge, ours is the first effort to apply ideas from FCA to a concept-based learning setting in vision to overlay a structured organization of concepts on a neural network's representations, which provides a strong semantic interpretation including at intermediate levels of a network.

3. Lattices for Concept-Based Learning

3.1. Background and Preliminaries

Concept-Based Models: We follow the standard CBM setup introduced by (Koh et al., 2020) and define a concept-based model as one that learns a mapping from $X \mapsto Y$ via an intermediate concept encoder $q(\cdot)$. Such models learn from a three-tuple dataset $\mathcal{D} = \{X, C, Y\}$ where $X \in \mathbb{R}^m$, $C \in \mathbb{R}^k$, $Y \in \mathbb{R}^n$ and m, k, n are dimensions of the image, concept and label spaces respectively. Each prediction is of the form $\hat{y} = p(q(x))$ where $q\colon X \mapsto C$ (e.g. *bird* image $\rightarrow$ {*white body*, *flat yellow bill*, ..., *orange legs*}) is the concept encoder, and $p\colon C \mapsto Y$ (e.g. {*white body*, *flat yellow bill*, ..., *orange legs*} $\rightarrow$ *Duck*) is an interpretable classifier network. These text-based concepts correspond to attributes (as discussed in earlier sections), and we refer to them as such henceforth.

Formal Concept Analysis (FCA): A *formal context* is defined as a three-tuple $\langle G, M, I\rangle$, where G is a set of objects, M is a set of attributes and $I \subseteq G \times M$ captures the binary relations (also called incidence relation) indicating which attributes are present in which objects. Given such a formal context, a *formal concept* (Ganter & Wille, 2024) is defined as a tuple $\langle A, B\rangle$, where A (*extent*) is a subset of objects and B (*intent*) is a subset of attributes. Note that these are not arbitrary subsets; these subsets of objects and attributes have concept-forming operators defined over them ($\uparrow, \downarrow$). A contains objects (classes in our case) sharing all attributes in B and B contains attributes shared by all objects in A. Given $A \subseteq G, B \subseteq M$, this is defined as:

$$A^{\uparrow} = B, B^{\downarrow} = A \tag{1}$$
$$A^{\uparrow} = \{m \in M \,|\, \forall g \in A : \langle x, y\rangle \in I\}$$
$$B^{\downarrow} = \{g \in G \,|\, \forall m \in B : \langle x, y\rangle \in I\}$$

The set of all formal concepts derived from a formal context forms a partial order over the subset-superset ordering relation, i.e. if $\langle A_1, B_1\rangle$, $\langle A_2, B_2\rangle$ are two formal concepts, $\langle A_1, B_1\rangle \preceq \langle A_2, B_2\rangle$ if $A_1 \subseteq A_2$ and $B_1 \supseteq B_2$, where $\preceq$ represents subconcept-superconcept ordering. This implies that general concepts have lesser attributes and more objects, while specific concepts have more attributes and lesser objects. For example, $\langle$*{apple, orange, carrot}*, *{food, edible}*$\rangle$ is more general than $\langle$*{apple}*, *{food, edible, sweet, fruit, red, round}*$\rangle$. This partial order allows us to construct a lattice of formal concepts, a hierarchy wherein concepts in higher layers are more general and ones in lower layers are more specific.

Definition 3.1 (Formal Concept Lattice)**.** Let $\mathcal{B}(G, M, I)$ denote the collection of all formal concepts of the formal context $\langle G, M, I\rangle$, i.e. $\mathcal{B}(G, M, I) = \{\langle A, B\rangle \in 2^G \times 2^M | A^{\uparrow} = B, B^{\downarrow} = A\}$. $\langle \mathcal{B}(G, M, I), \preceq\rangle$ is then a formal concept lattice (or simply lattice, in this work), where $\preceq$ is the subset-superset ordering. Let $\mathcal{L}$ denote this lattice, where $\mathcal{L} = \{\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_L\}$. $\mathcal{L}_i$ represents the set of formal concepts at level i, with $i = 1$ being the most general (top) level and $i = L$ the most specific.

3.2. Constructing Formal Concepts for Neural Networks

FCA (Ganter & Wille, 2024) provides a principled framework for organizing objects and attributes into a hierarchical structure based on their binary incidence relation. This naturally translates to a concept-based setting in DNNs where we have access to two interpretable sets: classes and their attributes. Treating classes as objects (G) and attributes as their properties (M), we define a formal context $\langle G, M, I\rangle$, where I is a binary relation indicating which classes contain which attributes. From this formal context, we construct a formal concept lattice: structured tuples of class and attribute subsets organized hierarchically (see Appendix for examples of formal concepts).

Fig 2 (top, in white box) shows an illustrative example of a lattice. The formal concepts shown at the bottom of the lattice contain singleton classes and the set of attributes that they contain. As we go up the lattice, the formal concepts become more general with increasing class set sizes and the corresponding maximal set of attributes they have in common. Attributes individually are not general or specific; it is the set of attributes that capture levels of generality. Note that the hierarchy herein captures a subset-superset ordering: attribute sets shared by more classes are more general, while those shared by fewer are more specific. We construct such a lattice for each dataset in our experiments; more lattice construction details are provided in the Appendix.

Extracting Attribute and Class Sets. The constructed formal concept lattice encodes multi-level semantic relationships, allowing us to extract supervision signals at varying levels of generality. We begin by organizing attributes into sets on the basis of their level in the lattice. Specifically, for each $\mathcal{L}_i$, we identify the set of formal concepts residing at that level. To represent attributes corresponding to a certain semantic level of generality, we compute the union of *intents* (i.e., sets of attributes) associated with formal concepts in $\mathcal{L}_i$:

$$M_i = \bigcup_{\texttt{fc} \in \mathcal{L}_i} \texttt{fc.intent} \tag{2}$$

where `fc.intent` and `fc.extent` denote the *intent* and *extent* of the formal concept. By repeating this process across different levels of the lattice, we obtain a hierarchy of attribute sets $\{M_1, M_2, \ldots, M_l\}$, each progressively more specific than the previous.

The concept lattice not only provides hierarchical attributes but also class groupings within each formal concept. We leverage this structure as a form of *supervision via iterative refinement*: we use loss functions at different layers such that they progressively narrow down the set of plausible classes (details in Sec 3.3). At early layers, general attribute

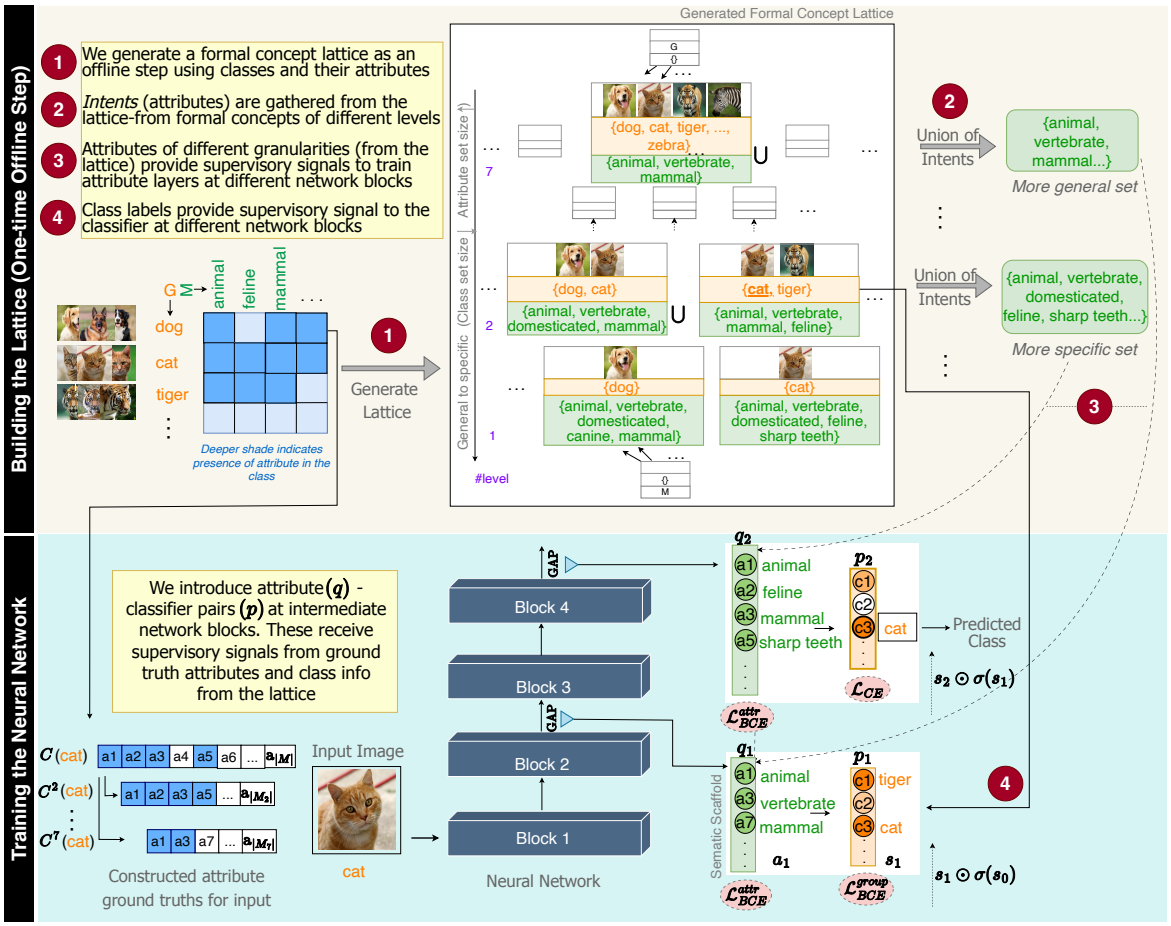


Figure 2. **Illustration of our overall approach.** A formal concept lattice is constructed for a concept-based setting using class and attribute sets and the binary relation between them *(top)*. A neural network is supervised at intermediate blocks using the information extracted from specific levels in the lattice *(bottom)*.

sets may not identify individual classes but can often disambiguate groups of classes. For example, the presence of attributes such as {*whiskers, fur*} can rule out classes like *tortoise* or *whale*, narrowing the prediction space from all classes to, say, {*cat, dog, lion, tiger*}. This class group is then *cascaded* forward, where subsequent layers refine it using more specific attributes; the next layer may use *domesticated* to narrow this down to {*cat, dog*}.

Such a learning mechanism allows suppression of classes that get eliminated early, focusing the network's discriminative capacity on the remaining candidates. We theoretically show in Sec 4 that this iterative refinement mechanism preserves the semantic ordering imposed by the lattice structure, minimizing ordering violations during training (Sec 3.3).

Class-Cluster Density. We formalize the notion of *class-cluster density* to quantify the semantic granularity of intermediate network representations and to align lattice levels with network depth. Let $f_j(x) \in \mathbb{R}^{d_j}$ denote the feature embedding of input x at network block j. Given a dataset with n classes, we apply k-means clustering with $k = n$ to the set $\{f_j(x)\}$ over the training samples. For a cluster K, let $\mathcal{Y}(K)$ denote the set of ground-truth class labels of samples assigned to K. We define the class-cluster density at block j as:

$$D_j = \frac{1}{n} \sum_{k=1}^{n} |\{\mathcal{Y}(K_k)\}| \tag{3}$$

where $\{K_k\}_{k=1}^{n}$ are the resulting clusters. Intuitively, D_j measures the average number of distinct classes grouped together by the representation at depth j; higher values indicate more class-agnostic (general) features, while lower values reflect increased class-specific separation.

An analogous quantity can be defined for the formal concept lattice. For each lattice level L_i, , we define this quantity as $\bar{D}_i = \frac{1}{|L_i|} \sum_{fc \in L_i} |\{\texttt{fc.extent}\}|$. By construction, higher lattice levels correspond to more general concepts and thus larger average extents, while lower levels correspond to finer-grained concepts with smaller extents. This establishes a natural alignment principle: early network blocks with high D_j should be supervised by higher lattice levels with large $\bar{D}_i$, while deeper blocks with lower D_j

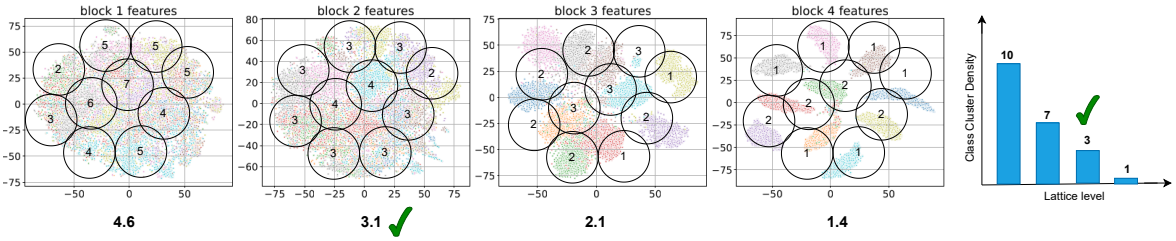


Figure 3. An illustrative example on 10 classes of how lattice levels are selected to supervise layers (blocks in our implementation) in the network. We perform clustering with n (here 10) centers at each layer and obtain the average number of unique classes present per cluster, compute the same over the formal concept extents at each lattice level, and then choose closest alignment.

should align with lower lattice levels. In practice, we assign each lattice level L_i to the earliest network block j such that $D_j \leq \bar{D}_i$, ensuring that semantic supervision is applied at a representational depth whose granularity matches that of the corresponding lattice level. This strategy is illustrated in Figure 3 and an algorithm is included in the Appendix.

3.3. FoCA CBM Training

We refer to our models as FoCA CBMs (Formal Concept Analysis CBMs). Once appropriate depths in the backbone have been determined, at each selected position j in the network, we introduce a concept encoder q_j to project the intermediate feature representation into a concept space defined by the attribute set M_i from lattice level i (i denotes an index in the lattice, j denotes an index in the network). Formally, each encoder learns a mapping $q_j : \mathbb{R}^{d_j} \to \mathbb{R}^{|M_i|}$ where d_j is the dimensionality of the downsampled feature at layer j, obtained via global average pooling, and $|M_i|$ is the size of the attribute set M_i (see Fig 2, bottom). We also introduce l classifiers $p_j : \mathbb{R}^{|M_i|} \to \mathbb{R}^n$, each operating on the output of the corresponding concept encoder q_j, where n is the total number of classes.

Obtaining Level-Wise Attribute and Class Ground Truths. Given a sample x with ground truth class label y, let $\mathbf{C}(x) \in \{0,1\}^{|M|}$ denote its complete ground truth attribute vector, where $|M|$ indicates all the attributes from the lattice. For lattice level i, let $\mathcal{M}_i \subseteq \{1,\ldots,|M|\}$ denote the set of attribute indices corresponding to attributes at that level. The level-wise ground truth attribute vector for sample x at level i is then obtained as $\mathbf{C}^i(x) = \mathbf{C}(x)[\mathcal{M}_i]$, i.e., by selecting the entries in $\mathbf{C}(x)$ indexed by attributes belonging to $\mathcal{M}_i$. Similarly, we define the *class group* of y at level i as the union of all class sets (extents) of formal concepts at $\mathcal{L}_i$ whose extents contain y. Formally, if $\mathcal{L}_i$ denotes the set of formal concepts at level i, then the class group is given by:

$$G_y^i = \bigcup_{\substack{\texttt{fc} \in \mathcal{L}_i \\ y \in \texttt{fc.extent}}} \texttt{fc.extent} \tag{4}$$

As an example, let level l_i contains the formal concepts $\langle\{dog, cat\}, \{animal, vertebrate, ...\}\rangle$, $\langle\{cat, tiger\}, \{animal, vertebrate, ...\}\rangle$ and $\langle\{tiger, zebra\}, \{animal, vertebrate, ...\}\rangle$. If sample x is a *cat*, since *cat* appears in two extents, the resulting group is $G_y^i = \{dog, cat, tiger\}$. $\bar{G}_y^i = \mathbf{1}_{[c \in G_{y_r}^{(i)}]}$ represents the corresponding binary vector. This is Step 4 in Fig 2. As we proceed to deeper lattice levels, these groups become progressively smaller, ultimately converging to singleton sets representing individual classes.

Overall Training Process. The network is jointly trained for both attribute prediction and classification, using the aforementioned iterative refinement strategy. For each attribute encoder output, we apply a binary cross-entropy loss ℓ_{BCE} against the corresponding attribute set. For all intermediate classifiers, we supervise using group-level labels derived from the lattice, again using ℓ_{BCE}. The final classifier is trained with standard cross-entropy loss ℓ_{CE} using the ground truth class label. Denoting $s_j(x) = p_j(q_j(f_j(x)))$ as the output for classes at layer j, we have $\hat{s}_j(x) = s_j(x) \cdot \sigma(s_{j-1}(x))$ as the post-iterative refinement output for classes at layer j, and $a_j(x) = \sigma(q_j(f_j(x)))$ the sigmoid output for attributes at layer j. The overall loss ℓ_{total} is hence:

$$\alpha \sum_{j=1}^{l} \underbrace{(\mathbf{C}^i(x) \cdot \log(a_j(x)) + (1 - \mathbf{C}^i(x) \cdot (1 - \log(a_j(x))))}_{\ell_{\text{BCE}_j}^{\text{attr}}}$$

$$+ \beta \sum_{j=1}^{l} \underbrace{(\bar{G}_y^i \cdot \log(\hat{s}_j(x)) + (1 - \bar{G}_y^i) \cdot (1 - log(\hat{s}_j(x)))}_{\ell_{\text{BCE}_j}^{\text{group}}}$$

$$+ \ell_{\text{CE}_l} \tag{5}$$

where α and β are weighting hyperparameters that balance the contribution of attribute and group-level supervision.

4. Some Theoretical Implications

We now present a theoretical analysis highlighting advantages of FoCA CBMs: (a) over non-hierarchical CBMs, and (b) more specifically, over other hierarchical CBM variants. Focusing first on (a), we study how imposing a lattice-ordered structure on classes and attributes provides a semantic scaffold for the network, in contrast to unordered alternatives, and show that FoCA CBMs preserve semantic concept ordering across layers. Let $f^{\texttt{foca}}$ denote a FoCA CBM trained with iterative refinement , and let $f^{\texttt{rnd}}$ denote

a network trained with random class groupings $\{G^i_{\text{rnd}}(y)\}$ that do not satisfy the subset constraint $G^{i+1}_{\text{rnd}}(y) \subseteq G^i_{\text{rnd}}(y)$. At zero training loss, preservation of ordering is trivial: a FoCA CBM respects lattice structure by construction, while random groupings may not. In general, multiple parameter configurations can attain the same empirical risk (Zhang et al., 2021). FoCA CBMs, however, introduce an inductive bias toward order-consistent solutions, ensuring that any formal concept activated at a given layer is also activated in all preceding layers, a guarantee absent under random group-based supervision. The substantive question lies in the realistic regime where the empirical risk $\hat{\ell}$ differs from the optimal risk ℓ^* by some margin $\epsilon > 0$. Crucially, not all prediction errors constitute ordering violations. For example, for a given input, if two consecutive layers incorrectly predict a true class as absent, the semantic ordering is preserved despite the misclassification. We therefore isolate and bound the probability of ordering-specific errors (configurations where $c \in \hat{G}_{i+1} \setminus \hat{G}_i$) as a function of the generalization gap $\epsilon = |\hat{\ell} - \ell^*|$. The following theorem establishes that, under bounded generalization error, a FoCA CBM maintains hierarchical consistency with high probability (over the training set X), while random supervision fails this guarantee even at zero training loss.

Theorem 4.1 (Inductive Bias towards Order Consistency)**.** *For an input $x \sim X$, define:*

$$\hat{G}_i(x) = \{g \in G \mid \hat{s}_j(x)[c] \geq \tau_c\},$$

where $\tau_c \in [0, 1]$ is a threshold, typically chosen as 0.5. Then under mild assumptions, given $|\widehat{\ell_{total}} - \ell^*_{total}| \leq \epsilon$,

$$\Pr_{f^{foca}}\left(\hat{G}_{i+1}(x) \not\subseteq \hat{G}_i(x)\right) \ll \Pr_{f^{rnd}}\left(\hat{G}_{i+1}(x) \not\subseteq \hat{G}_i(x)\right).$$

We provide a brief proof sketch herein, while deferring the detailed proof to the Appendix.

Proof Sketch. The proof proceeds in three steps. First, we establish that ground truth class groups respect the superset ordering: $G^{i+1}(y) \subseteq G^i(y)$ for all i and y. Second, we quantify the loss penalty for ordering violations. Each violation at transition $i \to i+1$ incurs excess loss of at least $\gamma_{\min} > 0$ compared to correct predictions. Third, we apply the asymmetric Lovász Local Lemma (Erdos & Lovász, 1975) to the collection of violation events $\{E_i\}_{i=1}^{L-1}$. The dependency structure where each event E_i depends only on adjacent events E_{i-1} and E_{i+1}, enables us to lower bound the probability of maintaining ordering across all layers as $\Pr(\bigcap_i \overline{E}_i)$. In stark contrast, random groupings yield high violation probabilities, via a simple combinatorial argument, even at zero training loss. □

Notably this affinity for order consistency holds vacuously for vanilla CBMs where mappings are restricted to only the most fine-grained bottom layer of a formal concept lattice. FoCA CBMs, on the other hand, generalize this approach by allowing mappings to any layer of the formal concept lattice. We next analyze the advantages of using a formal concept lattice as opposed to an arbitrary hierarchy. Drawing on insights from the information bottleneck theory (Tishby & Zaslavsky, 2015), we posit that the increase in mutual information between successive layers and the output $I(f_j(X); Y) - I(f_{j-1}(X); Y)$ is upper-bounded by a constant Δ, that depends on structural details such as the architecture and optimizer. In general, for DNNs, information gain between successive layers may be method-dependent and not guaranteed, as qualitatively illustrated by the t-SNE visualizations in the top row of Fig 4. In contrast, supervision of intermediate layers using a formal lattice induces a guaranteed lower bound on the information gain between any two consecutively supervised layers in FoCA-CBMs, a result formally established below in Theorem 4.2.

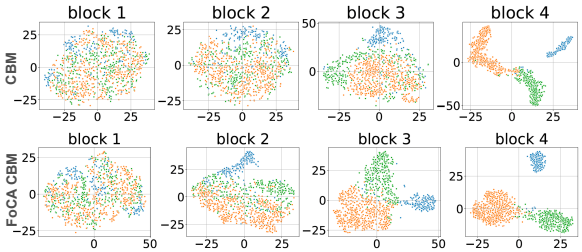


Figure 4. t-SNE plots of sample embeddings from AwA2 obtained from trained ResNet-18 backbones of Vanilla CBM and FoCA CBM. On a CBM, the clusters separate at the final block; in FoCA CBM, the separation happens gradually over blocks, showing graded semantic learning.

Theorem 4.2 (Information-Theoretic Benefit of FCA Supervision)**.** *Consider a FoCA-CBM trained with formal concept lattice $\mathcal{L}$ constructed from $\langle G, M, I\rangle$. Let network layer j be supervised by lattice level i via class groups $G^{(i)}$ and attribute sets M_i. Then, under bounded training error $|\hat{\ell} - \ell^*| \leq \epsilon$ with N training samples, the ϵ-calibrated information gain of the network for layer j is:*

$$I_{\mathcal{D}}(f_j(X); Y) - I_{\mathcal{D}}(f_{j-1}(X); Y) \geq \Delta^{(i)}_{\text{lattice}} - 2\Delta_{\text{align}}(\epsilon),$$

where $\Delta_{\text{align}}(\epsilon) = O\big(\sqrt{\epsilon \log |G|} + N^{-1/2}\big)$.

The formality property (Eqn 1) guarantees that each attribute set M_i is both informationally complete and parsimonious for its associated class-group structure. In contrast, random concept selection (Thm 4.1) breaks this optimality, resulting in ordering violations and unwarranted information loss.

5. Experiments

Datasets: We study our approach on three widely used benchmark datasets in concept-based learning: ImageNet100 (Russakovsky et al., 2015), AwA2 (Xian et al., 2019) and CIFAR100 (Krizhevsky et al., 2009). AwA2 is an expert-annotated dataset with class-level attributes, while for ImageNet100 and CIFAR100, we acquire class-level attribute annotations as in (Oikarinen et al., 2023) using

Table 1. Results on Classification Test Accuracy, Cluster Impurity (CI) and Cluster Compactness (DBI) on ImageNet100, AwA2, CIFAR100 datasets averaged over 3 seeds. Best in bold, second best underlined. FoCA CBM-N is a Naive variant of our method where the attribute sets obtained from the lattice are all stacked after the backbone followed by a standard classifier.

	ImageNet100			AwA2			CIFAR100		
	Acc ↑	CI ↓	DBI ↓	Acc ↑	CI ↓	DBI ↓	Acc ↑	CI ↓	DBI ↓
Vanilla CBM [ICML'20]	88.27±0.490	0.662±0.005	2.197±0.023	90.36±0.210	0.628±0.004	2.137±0.011	76.63±0.690	0.712±0.006	2.238±0.024
MLPCBM	86.88±0.290	0.659±0.006	2.210±0.029	89.65±0.370	0.637±0.007	2.120±0.018	76.17±0.620	0.722±0.006	2.276±0.027
Posthoc CBM [ICLR'23]	67.25±0.700	0.820±0.000	2.530±0.000	81.00±0.340	0.773±0.000	2.674±0.000	52.00±0.005	0.851±0.000	2.779±0.000
LFCBM [ICLR'23]	86.32±0.240	0.676±0.000	2.448±0.000	83.03±0.020	0.699±0.000	2.803±0.000	65.13±0.120	0.907±0.000	2.519±0.000
CEM [NeurIPS'22]	86.51±0.400	0.678±0.004	2.178±0.026	**93.11±0.170**	0.646±0.003	2.278±0.009	77.32±0.570	0.811±0.003	2.497±0.019
LaBo [CVPR'23]	74.48±0.004	0.676±0.000	2.448±0.000	91.53±0.010	0.699±0.000	2.803±0.000	65.23±0.003	0.907±0.000	2.519±0.000
SCBM [NeurIPS'24]	85.97±0.150	0.662±0.003	2.049±0.024	90.40±0.260	0.615±0.002	2.096±0.014	79.13±1.100	0.705±0.004	2.114±0.018
ProbCBM [ICML'23]	85.75±0.820	0.693±0.003	2.269±0.027	89.80±0.020	0.661±0.002	2.373±0.007	78.86±0.100	0.732±0.005	2.401±0.012
CF-CBM [NeurIPS'24]	87.30±0.010	0.676±0.000	2.448±0.000	89.19±0.810	0.699±0.000	2.803±0.000	60.02±0.540	0.907±0.000	2.519±0.000
HybridCBM [CVPR'25]	79.51±0.370	0.676±0.000	2.448±0.000	92.25±0.07	0.699±0.000	2.803±0.000	59.52±0.090	0.907±0.000	2.519±0.000
FoCA CBM-N [Ours]	88.36±0.290	0.665±0.005	2.150±0.021	88.26±0.270	0.659±0.005	2.273±0.015	**82.41±0.050**	0.688±0.005	2.177±0.021
FoCA CBM [Ours]	**91.88±0.350**	**0.573**±0.005	**1.862**±0.027	92.13±0.280	**0.571**±0.004	**2.057**±0.010	79.47±0.200	**0.622**±0.004	**1.855**±0.020

an LLM. We use these benchmark datasets as they capture conceptual diversity among them. More dataset details are in the Appendix.

Baselines: We compare our approach with ten concept-based learning models: (1) Vanilla CBMs (Koh et al., 2020), (2) MLPCBMs (an extension of vanilla CBMs), (3) Posthoc CBMs (Yuksekgonul et al., 2023), (4) Label-free CBMs (Oikarinen et al., 2023), (5) Concept Embedding Models (Espinosa Zarlenga et al., 2022), (6) Language in a Bottle (Yang et al., 2023), (7) Stochastic CBMs (Vandenhirtz et al., 2024a), (8) Probabilistic CBMs (Kim et al., 2023), (9) Coarse-to-Fine CBMs (Panousis et al., 2024), and (10) Hybrid CBMs (Liu et al., 2025). These models represent different flavors of concept-based approaches with strong performance. All models, including ours, are ResNet-based.

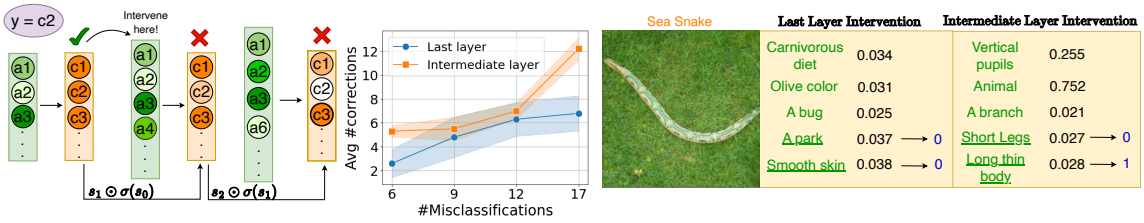
Metrics: In addition to classification accuracy, we evaluate the semantic quality of learned embeddings across network depth using clustering-based metrics. After training, we apply k-means clustering with n centers (= number of classes) to the feature embeddings at the end of each backbone block. We evaluate clusters using (i) Cluster Impurity (CI), using the Gini index (Breiman et al., 1984), which captures class heterogeneity within clusters, and (ii) Cluster Compactness, using the Davies–Bouldin Index (DBI) (Davies & Bouldin, 1979), which quantifies intra-cluster compactness and inter-cluster separation. Lower values of CI and DBI indicate more semantically structured representations.

Results: Our results over all metrics (*Test Accuracy, CI, DBI*) are reported for all baselines and our method in Table 1. FoCA CBM-N (N=Naive) is a naive variant of our method where attribute sets from multiple lattice levels are added as consecutive linear layers after the backbone. This is then trained like a Vanilla CBM with multiple attribute layers, followed by a classifier (one can view this as akin to a hierarchical CBM, with attribute sets constructed using our FCA lattice). The *CI* and *DBI* metrics give us a score per block of each model; we report the mean across all blocks. We see that our models consistently learn more meaningful embeddings, outperforming all baselines across datasets in terms of *CI* and *DBI*. In terms of accuracy, our models are consistently competitive across datasets.

Performance on ViT Backbones: Existing CBM methods largely focus on CNN architectures. To expand on this, we also studied our approach on ViT backbones, aligning appropriate levels from our lattice to intermediate blocks of a ViT architecture. We then compared our method (FoCA ViT) with a ViT CBM (CBM trained with a ViT backbone) on the CIFAR100 dataset. FoCA ViT outperformed the ViT CBM on all our metrics: 86.65 $\pm$ 0.295 vs 84.49 $\pm$ 0.547 test accuracy, 0.755 $\pm$ 0.004 vs 0.774 $\pm$ 0.016 CI and 1.983 $\pm$ 0.021 vs 2.237 $\pm$ 0.006 DBI, all averaged over three seeds.

6. Analysis

Considering Alternative Hierarchies: Our framework is not restricted to FCA-derived hierarchies. Any hierarchical structure satisfying the subset-superset ordering property can be used to guide network learning. To demonstrate this generality, we propose an alternative LLM-based hierarchy construction method: An LLM (GPT4) is provided with the class-attribute incidence matrix and is prompted to generate attribute sets that satisfy the subset-superset relation, along with a group of classes associated with each class. This method produces a hierarchy with the required ordering properties and can be integrated into our framework. Table 2 (top value in each row) shows the competitive performance of the GPT4-based model, demonstrating our method's generalizability. That said, we observe that FCA-based hierarchies are superior in terms of *concept leakage*. To quantify this, we randomly *turn off* some attributes (1%) in both models and measure the resulting drop in test accuracy (standard test to study concept leakage in CBMs). We



(a) (b) (c)

Figure 5. (a) Severity of misclassification informs which level of attributes to intervene on. Ground truth class for current input is $c2$, which is not in prediction set at layer 2; we hence intervene at 2nd layer; (b) Comparison of average number of corrections on interventions at last attribute layer (blue) and at appropriate intermediate layer on Imagenet100 (orange). Intervening at apt level improves performance; (c) Example attributes intervened on at last and intermediate layers; note that intermediate layers have more general attributes.

Table 2. Comparison with LLM (GPT4)-based hierarchy; both models show strong performance. However, turning off some attributes shows significant concept leakage in GPT4-based model, while FCA-based model shows strong concept compactness.

Hierarchy	ImageNet100	AwA2	CIFAR100
FCA	$91.88_{\pm 0.35}$ ↓ $60.57_{\pm 0.03}$	$92.13_{\pm 0.28}$ ↓ $91.51_{\pm 0.29}$	$79.47_{\pm 0.20}$ ↓ $51.06_{\pm 0.06}$
GPT4	$91.47_{\pm 0.41}$ ↓ $89.20_{\pm 0.40}$	$92.67_{\pm 0.10}$ ↓ $92.30_{\pm 0.39}$	$80.35_{\pm 0.07}$ ↓ $76.16_{\pm 0.08}$

observe that the model based on GPT4 hierarchy shows minimal change, indicating significant concept leakage, while our FCA-based model shows strong concept compactness.

Multi-Level Interventions: A key advantage of CBMs is their support for test-time interventions that probe concept-to-class mappings. While standard approaches apply random interventions at the final concept layer, our framework enables *level-aware* interventions aligned with semantic granularity. Intuitively, coarse misclassifications (e.g., *dog* vs. *elephant*) require intervention on general attributes, whereas fine-grained confusions (e.g., *dog* vs.*cat*) require more specific corrections. We quantify misclassification severity by identifying the deepest layer at which the ground-truth class remains in the predicted class group (Fig 5(a)). Due to iterative refinement, classes eliminated at a given level rarely reappear, making this layer a natural intervention point. We intervene at the corresponding attribute layer by randomly modifying k attributes and propagating the updated activations forward, and compare this to interventions applied only at the final layer. As shown in Fig 5(b), level-aware interventions consistently outperform final-layer interventions across four random sets of misclassifications on ImageNet100 (averaged over 10 trials of 20 interventions per sample). Fig 5(c) illustrates that earlier layers predominantly involve general attributes, whereas final-layer interventions mix general and fine-grained concepts. Together, these results show that semantic scaffolding enables effective interventions at multiple abstraction levels.

Scaling and Practicality: Formal concept lattices are constructed as a one-time offline preprocessing step, taking approximately 4.5 seconds for ImageNet100, 37 seconds for AwA2, and 2 seconds for CIFAR100. The worst-case complexity of lattice construction is $O(|\mathcal{L}| \cdot |G|^2 \cdot |M|)$, where $|\mathcal{L}|$ is the number of formal concepts; however, real-world class–attribute relations are typically sparse (fill ratio < 0.1), resulting in near-quadratic growth in practice (Table A11). Table 3 reports the computational cost (in FLOPs) of FoCA CBMs and representative baselines. Despite introducing intermediate supervision, FoCA CBMs incur computational costs comparable to standard CBMs. Prior work has also shown that formal concepts remain stable under incremental updates (Kuznetsov, 2007), supporting the applicability of FCA-based hierarchies in dynamic settings.

Table 3. Computational cost (in Giga Floating Point Operations) of different models on all three datasets.

Model	ImageNet100	AwA2	CIFAR100
CBM	8.21G	3.64G	8.21G
CEM	8.259G	3.64G	8.26G
SCBM	17.44G	7.28G	17.44G
FoCA CBM	8.21G	3.64G	8.21G

Further analysis, ablations, and more results including interventions, impact of iterative refinement, lattice and position choices are provided in the Appendix.

7. Conclusion

In this work, we examine how concept-based learning models can respect hierarchical structure across a network's depth. Rather than learning concepts as a flat set at a single stage, we propose guiding representation learning using a structured semantic hierarchy derived from Formal Concept Analysis. The resulting formal concept lattice provides principled supervision points that align semantic granularity with network depth, enabling representations to evolve naturally from general to specific. Our approach improves interpretability by exposing meaningful intermediate concepts and enhances learning through gradual semantic refinement, as reflected in improved clustering quality. We further show that multi-level interventions are both feasible and more effective than standard single-layer alternatives through our approach. We believe this framework brings concept-based models closer to human semantic reasoning, and opens avenues for weakly supervised hierarchies and extensions to other modalities.

Impact Statement

Our work advances the fields of Concept-Based Learning and Interpretability by establishing a principled method for organizing semantic concepts in neural networks. The hierarchical structure we introduce enables multi-level interpretability, where users can interact with models at varying levels of abstraction. We believe that this has significant implications for high-stakes domains like healthcare and medical diagnostics where the operators can verify that a model's reasoning respects the appropriate diagnostic hierarchies (identifying organs before specific pathologies). The multi-level intervention capability enables more nuanced interactions with these models. Users can now target corrections at the appropriate level of semantic granularity - a broad category mistake or finer-grained mistake. By formalizing the connection between Formal Concept Analysis and Deep Learning, our work bridges communities that have traditionally worked in isolation. We believe that this cross-pollination could inspire new research directions leading to models that better align with human cognitive structures.

References

Berend, D. and Kontorovich, A. A finite sample analysis of the naive bayes classifier. *Journal of Machine Learning Research*, 16(44):1519–1545, 2015. URL http://jmlr.org/papers/v16/berend15a.html.

Breiman, L., Friedman, J., Stone, C., and Olshen, R. *Classification and Regression Trees*. Taylor & Francis, 1984. ISBN 9780412048418. URL https://books.google.co.in/books?id=JwQx-WOmSyQC.

Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., and Su, J. K. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019.

Davies, D. L. and Bouldin, D. W. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979. doi: 10.1109/TPAMI.1979.4766909.

Durrschnabel, D., Hanika, T., and Stubbemann, M. Fca2vec: Embedding techniques for formal concept analysis. In *Complex Data Analytics with Formal Concept Analysis*, 2019. URL https://api.semanticscholar.org/CorpusID:208291462.

Erdos, P. and Lovász, L. Problems and results on 3-chromatic hypergraphs and some related questions. In *Infinite and finite sets (Colloq., Keszthely, 1973; dedicated to P. Erdős on his 60th birthday), Vols. I, II, III*, volume Vol. 10 of *Colloq. Math. Soc. János Bolyai*, pp. 609–627. North-Holland, Amsterdam-London, 1975.

Espinosa Zarlenga, M., Barbiero, P., Ciravegna, G., Marra, G., Giannini, F., Diligenti, M., Shams, Z., Precioso, F., Melacci, S., Weller, A., et al. Concept embedding models: Beyond the accuracy-explainability trade-off. *Advances in Neural Information Processing Systems*, 35:21400–21413, 2022.

Ganea, O., Becigneul, G., and Hofmann, T. Hyperbolic entailment cones for learning hierarchical embeddings. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1646–1655. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/ganea18a.html.

Ganter, B. and Wille, R. *Formal concept analysis: mathematical foundations*. Springer Nature, 2024.

Giunchiglia, E. and Lukasiewicz, T. Coherent hierarchical multi-label classification networks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Giunchiglia, E. and Lukasiewicz, T. Multi-label classification neural networks with hard logical constraints. *J. Artif. Int. Res.*, 72:759–818, January 2022. ISSN 1076-9757. doi: 10.1613/jair.1.12850. URL https://doi.org/10.1613/jair.1.12850.

Gyurek, C., Talukder, N., and Hasan, M. A. Binder: Hierarchical concept representation through order embedding of binary vectors. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 980–991, 2024.

Harvey, N. J. and Vondrák, J. An algorithmic proof of the lovasz local lemma via resampling oracles. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pp. 1327–1346, 2015. doi: 10.1109/FOCS.2015.85.

Kearns, M. and Saul, L. Large deviation methods for approximate probabilistic inference. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, pp. 311–319, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 155860555X.

Kim, E., Jung, D., Park, S., Kim, S., and Yoon, S.-H. Probabilistic concept bottleneck models. *ArXiv*, abs/2306.01574, 2023. URL https://api.semanticscholar.org/CorpusID:259063823.

Koh, P. W., Nguyen, T., Tang, Y. S., Mussmann, S., Pierson, E., Kim, B., and Liang, P. Concept bottleneck models. In *International conference on machine learning*, pp. 5338–5348. PMLR, 2020.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Kuznetsov, S. O. On stability of a formal concept. *Annals of Mathematics and Artificial Intelligence*, 49(1):101–115, 2007.

Li, L., Wang, W., and Yang, Y. Logicseg: Parsing visual semantics with neural logic learning and reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4122–4133, October 2023.

Liu, Y., Zhang, T., and Gu, S. Hybrid concept bottleneck models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 20179–20189, 2025.

Marconato, E., Passerini, A., and Teso, S. Glancenets: Interpretabile, leak-proof concept-based models. In *Neural Information Processing Systems*, 2022. URL https://api.semanticscholar.org/CorpusID:249209924.

Nickel, M. and Kiela, D. Poincaré embeddings for learning hierarchical representations. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/59dfa2df42d9e3d41f5b02bfc32229dd-Paper.pdf.

Oikarinen, T., Das, S., Nguyen, L. M., and Weng, T.-W. Label-free concept bottleneck models. *arXiv preprint arXiv:2304.06129*, 2023.

Olah, C., Satyanarayan, A., Johnson, I., Carter, S., Schubert, L., Ye, K., and Mordvintsev, A. The building blocks of interpretability. *Distill*, 3(3):e10, 2018.

Panousis, K. P., Ienco, D., and Marcos, D. Coarse-to-fine concept bottleneck models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=RMdnTnffou.

Peng, S., Yang, H., and Yamamoto, A. Bert4fca: A method for bipartite link prediction using formal concept analysis and bert. *Plos one*, 19(6):e0304858, 2024.

Raman, N. J., Espinosa Zarlenga, M., and Jamnik, M. Understanding inter-concept relationships in concept-based models. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 42009–42025. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/raman24a.html.

Rudolph, S. Using fca for encoding closure operators into neural networks. In *International Conference on Conceptual Structures*, pp. 321–332. Springer, 2007.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Sarkar, A., Vijaykeerthy, D., Sarkar, A., and Balasubramanian, V. N. A framework for learning ante-hoc explainable models via concepts. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10276–10285, 2022.

Sawada, Y. and Nakamura, K. Concept bottleneck model with additional unsupervised concepts. *IEEE Access*, 10:41758–41765, 2022. doi: 10.1109/ACCESS.2022.3167702.

Sinha, S., Huai, M., Sun, J., and Zhang, A. Understanding and enhancing robustness of concept-based models. In *AAAI Conference on Artificial Intelligence*, 2022. URL https://api.semanticscholar.org/CorpusID:254069697.

Srivastava, D., Yan, G., and Weng, T.-W. Vlg-cbm: Training concept bottleneck models with vision-language guidance. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 79057–79094. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/90043ebd68500f9efe84fedf860a64f3-Paper-Conference.pdf.

Sun, A., Yuan, Y., Ma, P., and Wang, S. Eliminating information leakage in hard concept bottleneck models with supervised, hierarchical concept learning. *arXiv preprint arXiv:2402.05945*, 2024.

Theves, S., Neville, D. A., Fernández, G., and Doeller, C. F. Learning and representation of hierarchical concepts in hippocampus and prefrontal cortex. *Journal of Neuroscience*, 41(36):7675–7686, 2021.

Tishby, N. and Zaslavsky, N. Deep learning and the information bottleneck principle, 2015. URL https://arxiv.org/abs/1503.02406.

Troy, A. D., Zhang, G.-Q., and Tian, Y. Faster concept analysis. In *Proceedings of the 15th International Conference on Conceptual Structures: Knowledge Architectures for Smart Applications*, ICCS '07, pp. 206–219, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 9783540736806. doi: 10.1007/978-3-540-73681-3_16. URL https://doi.org/10.1007/978-3-540-73681-3_16.

Vandenhirtz, M., Laguna, S., Marcinkevičs, R., and Vogt, J. Stochastic concept bottleneck models. *Advances in Neural Information Processing Systems*, 37:51787–51810, 2024a.

Vandenhirtz, M., Laguna, S., Marcinkevičs, R., and Vogt, J. Stochastic concept bottleneck models. *Advances in Neural Information Processing Systems*, 37:51787–51810, 2024b.

Vendrov, I., Kiros, R., Fidler, S., and Urtasun, R. Order-embeddings of images and language. *arXiv preprint arXiv:1511.06361*, 2015.

Xian, Y., Lampert, C. H., Schiele, B., and Akata, Z. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE Transactions on Pattern Analysis &; Machine Intelligence*, 41(09):2251–2265, sep 2019. ISSN 1939-3539. doi: 10.1109/TPAMI.2018.2857768.

Yang, Y., Panagopoulou, A., Zhou, S., Jin, D., Callison-Burch, C., and Yatskar, M. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19187–19197, 2023.

Yuksekgonul, M., Wang, M., and Zou, J. Post-hoc concept bottleneck models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=nA5AZ8CEyow.

Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pp. 818–833. Springer, 2014.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM*, 64(3):107–115, February 2021. ISSN 0001-0782. doi: 10.1145/3446776. URL https://doi.org/10.1145/3446776.

Appendix

In this appendix, we provide additional details of our work, including the following information.

Table of Contents

A1. Notation

Symbol	Description
$\langle G, M, I \rangle$	Formal context of the set of objects G, the set of attributes M, with the incidence relation I
$\langle A, B \rangle$	Formal concept with the extent $A \subseteq G$ and the intent $B \subseteq M$
$\uparrow, \downarrow$	Concept-forming operators
$\preceq$	Subset-superset ordering
$\mathcal{B}(G, M, I)$	Set of all formal concepts corresponding to the formal context $\langle G, M, I \rangle$
$\mathcal{L}$	Formal concept lattice, $\mathcal{L} = \langle \mathcal{B}(G, M, I), \preceq \rangle$
L	Number of levels in the lattice
$\mathcal{L}_1, \ldots, \mathcal{L}_L$	Set of formal concepts at each level of the lattice
$M_1, \ldots, M_L$	Attribute set at each level of the lattice
n	Total number of classes
σ	Sigmoid
d_j	Dimensionality of the downsampled feature space at selected position j in the backbone network
f_j	Global average pooling output at semantic layer j for input x, $f_j : \mathcal{X} \to \mathbb{R}^{d_j}$
q_j	Concept encoder for semantic layer j corresponding to lattice layer i, $q_j : \mathbb{R}^{d_j} \to \mathbb{R}^{\lvert M_i \rvert}$
p_j	Classifier for semantic layer j corresponding to lattice layer i, $p_j : \mathbb{R}^{\lvert M_i \rvert} \to \mathbb{R}^n$
s_j	Classifier output at semantic layer j for input x, $s_j : \mathcal{X} \to \mathbb{R}^n = p_j \circ q_j \circ f_j$
$\hat{s}_j$	Post-iterative refinement output at semantic layer j for input x, $\hat{s}_j : \mathcal{X} \to \mathbb{R}^n = s_j \cdot (\sigma \circ s_{j-1})$
a_j	Attribute sigmoid output at semantic layer j for input x, $a_j : \mathcal{X} \to \mathbb{R}^n = \sigma \circ q_j \circ f_j$
$\mathbf{C}$	Complete ground truth attribute vector for sample x
$\mathbf{C}^i$	Ground truth attribute vector for sample x at lattice level i
G_y^i	Class group of sample x with class label y at lattice level i
$H(\cdot)$	Entropy
$I_{\mathcal{D}}(\cdot;\cdot)$	Mutual information for dataset $\mathcal{D}$

A2. Ablations

Ablation on Iterative Refinement: We study the impact of *iterative refinement* and perform an ablation with and without it on our method. We see a consistent drop in accuracy across datasets without iterative refinement, with consistent increases in *CI* and *DBI* as well, on almost all datasets. We report these results in Tables A4, A5 and A6. These results highlight the usefulness of our class group-based refinement strategy.

Table A4. Our group-based iterative refinement strategy improves on all metrics on CIFAR100.

	CIFAR100		
	Acc ↑	CI ↓	DBI ↓
With	79.7	0.6221	1.8558
Without	77.19	0.6612	1.8762

Table A5. Our group-based iterative refinement strategy improves on all metrics on ImageNet100.

	ImageNet100		
	Acc ↑	CI ↓	DBI ↓
With	91.92	0.5737	1.8623
Without	88.19	0.6162	2.0639

Ablation on Class-Cluster Density: We study the impact of class-cluster-density-based backbone position selection and lattice level selection. In order to isolate the impacts, we try both (1) fixing the backbone position and using non class-cluster-density-based lattice level selection, and (2) fixing the lattice level and using non class-cluster-density-based

Table A6. Our group-based iterative refinement strategy improves on almost all metrics on AwA2, with the CI scores being in the same range.

	AwA2		
	Acc ↑	CI ↓	DBI ↓
With	92.20	0.5288	1.932
Without	91.85	0.5231	2.017

backbone position selection. We use two semantic layers for these experiments and provide the results in Tables **??** and **??**. The results indicate how different levels carry different amounts of information.

Table A7. Accuracy on varying the semantic layers position in FoCA CBM models on all datasets. Models have the following naming scheme: `FoCA_<levels>_<positions>`. For example, FoCA_(2, 1)_(3, 4) indicates that the attributes are extracted from the 2nd and 1st levels in the lattice, and are used to supervise the 3rd and 4th blocks in the ResNet. clf0 and clf1 correspond to the two classifiers from the two semantic layers.

Model	AwA2		Model	CIFAR100		Model	ImageNet100	
	clf0	clf1		clf0	clf1		clf0	clf1
FoCA_(2, 1)_(2, 4)	85.61	89.88	FoCA_(2, 1)_(2, 4)	92.41	68.89	FoCA_(5,1)_(2, 4)	79.56	90.69
FoCA_(2, 1)_(3, 4)	87.90	92.20	FoCA_(2, 1)_(3, 4)	94.30	79.70	FoCA_(5,1)_(3, 4)	96.26	91.92

Table A8. Accuracy on varying the levels chosen from the lattice in FoCA CBM models on all datasets.Models have the following naming scheme: `FoCA_<levels>_<positions>`. For example, FoCA_(3, 1)_(3, 4) indicates that the attributes are extracted from the 3rd and 1st levels in the lattice, and are used to supervise the 3rd and 4th blocks in the ResNet. clf0 and clf1 correspond to the two classifiers from the two semantic layers.

Model	AwA2		Model	CIFAR100		Model	ImageNet100	
	clf0	clf1		clf0	clf1		clf0	clf1
FoCA_(3, 1)_(3, 4)	87.90	92.20	FoCA_(2, 1)_(3, 4)	94.30	79.70	FoCA_(3,1)_(3, 4)	91.80	88.30
FoCA_(4, 1)_(3, 4)	95.40	91.61	FoCA_(3, 1)_(3, 4)	92.16	78.44	FoCA_(5,1)_(3, 4)	96.26	91.92
FoCA_(6, 1)_(3, 4)	98.27	91.77	FoCA_(5, 1)_(3, 4)	89.26	73.91	FoCA_(7,1)_(3, 4)	93.98	91.00

A3. Proofs and Further Theoretical Analysis

In this section, we provide the proof of Theorem 1, along with a similar theorem for attributes.

Theorem 4.1 (Inductive Bias towards Order Consistency)**.** *For an input $x \sim X$, define:*

$$\hat{G}_i(x) = \{g \in G \mid \hat{s}_j(x)[c] \geq \tau_c\},$$

where $\tau_c \in [0, 1]$ is a threshold, typically chosen as 0.5. Then under mild assumptions, given $|\widehat{\ell_{\texttt{total}}} - \ell^*_{\texttt{total}}| \leq \epsilon$,

$$\Pr_{f^{\texttt{foca}}}\left(\hat{G}_{i+1}(x) \not\subseteq \hat{G}_i(x)\right) \ll \Pr_{f^{\texttt{rnd}}}\left(\hat{G}_{i+1}(x) \not\subseteq \hat{G}_i(x)\right).$$

Proof. We provide mathematical details to the proof sketch described in Section 4 of the main paper. We begin by proving the following simple lemma.

Step 1: Ground Truth Ordering.

Lemma A3.1. *$G^{i+1}(y) \subseteq G^i(y)$ for all $i \in [L-1]$, $y \in G$.*

Proof. For sake of contradiction, suppose $\exists c \in G^{i+1}(y) \setminus G^i(y)$. Then $\exists \langle A, B\rangle \in \mathcal{L}_{i+1}$ with $c, y \in A$. By lattice structure, $\exists \langle C, D\rangle \in \mathcal{L}_i$ with $A \subseteq C$ (subconcept property). Thus $c \in C$ and $y \in C$, implying $C \subseteq G^i(y)$ by definition, giving $c \in G^i(y)$, which is absurd. Hence, the assertion in the lemma holds. □

Algorithm 1 Training Lattice-Guided Concept-Based Models

Require: Dataset $\mathcal{D} = \{(x, c, y)\}$, class-attribute annotations (G, M, I), base network f, number of supervision points l, loss weights α, β

$\mathcal{L} \leftarrow \textsc{ConstructLattice}(G, M, I)$ ▷ Build concept lattice
for $i = 1$ to l **do**
  $\mathcal{M}_i \leftarrow \bigcup_{\langle A,B\rangle \in \mathcal{L}_i} B$ ▷ Extract attribute set from level i of the lattice $\mathcal{L}$
end for
$\{j_1, \ldots, j_l\}, \{\mathcal{L}_1, \ldots, \mathcal{L}_l\} \leftarrow \textsc{SelectLayersAndLevels}(f, \mathcal{D}, \mathcal{L}, l)$ ▷ Select supervision points in network and lattice levels using class cluster density
Initialize concept encoders $\{q_1, \ldots, q_l\}$ and classifiers $\{p_1, \ldots, p_l\}$
for epoch $= 1$ to N **do**
  for each $(x, c, y) \in \mathcal{D}$ **do**
    $\hat{y}_0 \leftarrow \infty$ ▷ Initialize with all classes possible
    $\ell_{\text{attr}} \leftarrow 0$, $\ell_{\text{group}} \leftarrow 0$, $\ell_{\text{class}} \leftarrow 0$
    for $i = 1$ to l **do**
      $h_i \leftarrow f_{\leq j_i}(x)$ ▷ Forward through layer j_i
      $\hat{c}_i \leftarrow q_i(h_i)$ ▷ Predict concepts from M_i
      $\hat{y}_i \leftarrow p_i(\hat{c}_i)$ ▷ Predict classes
      $C_y^i \leftarrow [c[\mathcal{M}_i]]$ ▷ Ground truth concepts for level i by selecting the concepts from c that are present in $\mathcal{M}_i$
      $G_y^i \leftarrow \bigcup_{\langle A,B\rangle \in \mathcal{L}_i, y \in A} A$ ▷ Ground truth group
      $\ell_{\text{attr}} \leftarrow \ell_{\text{attr}} + \text{BCE}(\sigma(\hat{c}_i), C_y^i)$ ▷ Concept loss
      $\hat{y}_i \leftarrow \hat{y}_i \odot \sigma(\hat{y}_{i-1})$ ▷ Iterative refinement
      if $i < l$ **then**
        $\ell_{\text{group}} \leftarrow \ell_{\text{group}} + \text{BCE}(\sigma(\hat{y}_i), G_y^i)$ ▷ Group loss
      else
        $\ell_{\text{class}} \leftarrow \text{CE}(\textsc{softmax}(\hat{y}_i), y)$ ▷ Final class loss
      end if
    end for
    $\ell_{\text{total}} \leftarrow \alpha \cdot \ell_{\text{attr}} + \beta \cdot \ell_{\text{group}} + \ell_{\text{class}}$
    Update parameters via backpropagation on ℓ_{total}
  end for
end for
return Trained model with hierarchical concept structure

Step 2: Loss Penalty for Violations.

Next, we examine how much extra loss is incurred by an ordering mismatch over a prediction that obeys the ground truth, to later help us derive a bound on the probability of ordering mismatches. For simplicity, we assume the $0-1$ loss, and we only consider the group loss for the rest of this proof. First, note that for a sample (x, y) and class c, the per-layer loss is:

$$\mathcal{L}_i^c = \begin{cases} -\log \hat{y}_i^c & \text{if } c \in G^i(y) \\ -\log(1 - \hat{y}_i^c) & \text{if } c \notin G^i(y) \end{cases} \tag{6}$$

Lemma A3.2. *If $c \in \mathcal{C}_{i+1} \setminus \mathcal{C}_i$ (ordering violation), then:*

$$\mathcal{L}_i^c + \mathcal{L}_{i+1}^c \geq \mathcal{L}_i^{c,*} + \mathcal{L}_{i+1}^{c,*} + \gamma_{\min} \tag{7}$$

where $\mathcal{L}_i^{c,*}$ is the optimal loss achievable respecting ordering, and $\gamma_{\min} = 1$.

Proof. Since neural networks are universal function approximators, we take $\mathcal{L}_i^{c,*} = 0$. By Lemma A3.1, there are three exhaustive cases to consider, namely:

- *Case 1:* $c \in G^{i+1}(y) \subseteq G^i(y)$. Ground truth requires $\hat{y}_i^c, \hat{y}_{i+1}^c \geq \tau$.

- *Case 2:* $c \in G^i(y) \setminus G^{i+1}(y)$. Ground truth requires $\hat{y}_i^c \geq \tau$, $\hat{y}_{i+1}^c < \tau$.
- *Case 3:* $c \notin Y^i(y) \cup Y^{i+1}(y)$. Ground truth requires both predictions low.

Simple casework in each of these cases gives $\gamma_{\min} = 1$.

□

Step 3: Probability Upper Bound For a Single Ordering Mismatch.

Let E_i^c denote the event $c \in \mathcal{C}_{i+1} \setminus \mathcal{C}_i$. By the definition of the group loss function and its contribution to the total loss (Objective 5), we know that the risk $\widehat{\ell_{\texttt{group}}} \leq \widehat{\ell_{\texttt{total}}}$. Furthermore, we assume that $\ell^*_{\texttt{total}} = \ell^*_{\texttt{group}} = 0$ owing to the universal function approximation capabilities of neural networks. Therefore given that $|\widehat{\ell_{\texttt{total}}} - \ell_{\texttt{total}}| \leq \epsilon$, we can telescope individual risks specific to layer indices and classes to write,

$$|\widehat{\ell_{\texttt{total}}} - \ell_{\texttt{total}}| \leq \epsilon \iff \textstyle\sum_{i=1}^{L-1} \sum_{c \in G} \mathcal{L}_i^c + \mathcal{L}_{i+1}^c \leq \sum_{i=1}^{L-1} \sum_{c \in G} \mathcal{L}_i^{c,*} + \mathcal{L}_{i+1}^{c,*} + 2\epsilon.$$

Then by Lemma A3.2 and the definition of the loss function, $\ell_{\texttt{group}}$,

$$\sum_{i=1}^{L-1} \sum_{c \in G} \mathbb{E}_{(x,y)\sim\mathcal{D}}[E_i^c \cdot \gamma_{\min}] \leq 2\epsilon. \tag{8}$$

Since E_i^c is a Bernoulli random variable, we can equivalently write

$$\sum_{i=1}^{L-1} \sum_{c \in G} \mathtt{P}(E_i^c) \leq \frac{2\epsilon}{\gamma_{\min}}. \tag{9}$$

Step 4: Applying Asymmetric Lovász Local Lemma.

To provide a tighter lower bound on the probability that no ordering mismatch occurs under lighter assumptions, we turn to applying the (asymmetric version of the) Lovász Local Lemma (*LLL*) to the events E_i^c. We begin by stating the *LLL* as follows (Harvey & Vondrák, 2015; Erdos & Lovász, 1975):

Lemma A3.3 (*Asymmetric Lovász Local Lemma*)**.** *Let $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_n$ be events in a probability space with a dependency graph such that $\mathtt{P}(\mathcal{E}_i) \leq p_i \in [0,1]$. If $\exists y_1, y_2, \ldots y_n > 0$ for which:*

$$p_i \leq \frac{y_i}{\sum_{S \subseteq \Gamma^+(i)} \Pi_{j \in S}(1 + y_j)}, \ \forall i \in [n], \tag{10}$$

then $\mathtt{P}(\bigcap_{i=1}^n \overline{\mathcal{E}_i}) = \Pi_{i \in [n]} \frac{1}{(1+y_i)} > 0$.

Here, $\Gamma^+(i)$ denotes the union of the neighbors of the event $\mathcal{E}_i$ (along with itself) in the dependency graph that exists over the events. In our setting, the only neighbour that a given event $\mathcal{E}_i^c$ directly depends on is $\mathcal{E}_{i-1}^c$ for $i > 0$. Therefore applying the *LLL* in our setting reduces to finding $y_1, y_2, \ldots y_{|G| \times L}$ such that,

$$\Pr(E_i^c) \leq \frac{y_{\{c,i\}}}{(1 + y_{\{c,i\}})(1 + y_{\{c,i-1\}})}, \ \forall i \in [n] \setminus \{0\}, c \in G. \tag{11}$$

To do so, we first note that since the events E_i^c correspond to Bernoulli random variables, we can apply a tight inequality applicable for Bernoulli random variables known as the Kearns-Saul inequality (Kearns & Saul, 1998; Berend & Kontorovich, 2015), stated as follows:

Lemma A3.4 (*Kearns-Saul Inequality*)**.** *For all $p \in (0,1)$ and $t \in \mathbb{R}$,*

$$(1-p)e^{-tp} + pe^{t(1-p)} \leq e^{\left(\frac{1-2p}{4\log\left(\frac{1-p}{p}\right)} t^2\right)}. \tag{12}$$

The above inequality can be rewritten as,

$$p \leq \frac{e^{\left(\frac{1-2p}{4\log\left(\frac{1-p}{p}\right)}\right)t^2+pt} - 1}{(e^t - 1)}, \ \forall p \in (0,1), t \in \mathbb{R}. \tag{13}$$

Notice that $f(t) = \left(\frac{1-2p}{4\log\left(\frac{1-p}{p}\right)}\right)t^2 + pt - t$ attains its minimum value at $t^*(p) = 2 \cdot \frac{1-p}{1-2p}\log(\frac{1-p}{p})$. Furthermore, $f(t^*(p)) \leq 0$. Therefore for a fixed $p \in (0,1)$ we can write,

$$p \leq \frac{e^{f(t^*(p))+t^*(p)} - 1}{(e^{t^*(p)} - 1)} \leq e^{f(t^*(p))}. \tag{14}$$

Now let $\kappa = 1.35$ be a fixed constant. Choose $y_{\{c,i\}} = e^{\kappa P(E_i^c)} - 1$. Notice that,

$$\frac{y_{\{c,i\}}}{(1+y_{\{c,i\}})(1+y_{\{c,i-1\}})} \tag{15}$$

$$= e^{-\kappa(\mathtt{P}(E_{i-1}^c)+\mathtt{P}(E_i^c))} \cdot (e^{\kappa \mathtt{P}(E_i^c)} - 1) \tag{16}$$

$$\geq e^{-\kappa(\mathtt{P}(E_{i-1}^c)+\mathtt{P}(E_i^c))} \cdot (\mathtt{P}(E_i^c) e^{\kappa^{-0.01}}), \tag{17}$$

where the last inequality can be obtained by simple differentiation and checking for sign changes at various values of κ.

By using our assumption that $\mathtt{P}(E_i^c) + \mathtt{P}(E_{i-1}^c) \leq 0.67$, we have that,

$$\kappa^{-0.01} - \kappa(\mathtt{P}(E_i^c) + \mathtt{P}(E_{i-1}^c)) \geq 0.08388 \tag{18}$$

$$= \max_{\mathtt{P}(E_i^c)} f(t^*(\mathtt{P}(E_i^c))) - \log(\mathtt{P}(E_i^c))), \tag{19}$$

where the last inequality again follows from techniques based on simple differentiation and checking of sign values at relevant points.

Therefore,

$$\mathtt{P}(E_i^c) \leq \frac{y_{\{c,i\}}}{(1+y_{\{c,i\}})(1+y_{\{c,i-1\}})}, \ \forall i \in [n] \setminus \{0\}, c \in G. \tag{20}$$

Therefore, $\mathtt{P}(\bigcap_{i=1}^{|G|\cdot L} \overline{\mathcal{E}_i}) = \Pi_{i \in [|G|\cdot L]} \frac{1}{(1+y_i)} \geq \Pi_{i \in [|G|\cdot L]} e^{-\kappa \mathtt{P}(E_i^c)} \geq e^{-\frac{2\epsilon\kappa}{\gamma_{\min}}}$. □

Step 5: Random Baseline Analysis.

Lemma A3.5. *For random groupings where $G_{rnd}^{i+1}(y) \not\subseteq G_{rnd}^{i}(y)$ in general, with group sizes k_{i+1}, k_i:*

$$\Pr(E_i^c) \geq \frac{k_{i+1}}{|G|}\left(1 - \frac{k_i}{|G|}\right) \tag{21}$$

For hierarchies with $k_i = \Theta(|G|/L)$, this gives $\Pr(E_i^c) = \Omega(1/L)$.

Proof. Under random independent assignment, a class c satisfies:

$$\Pr(c \in G_{\text{rnd}}^{i+1}(y)) = \frac{k_{i+1}}{|G|} \tag{22}$$

$$\Pr(c \notin G_{\text{rnd}}^{i}(y)) = 1 - \frac{k_i}{|G|} \tag{23}$$

Therefore:

$$\Pr(c \in G^{i+1}_{\text{rnd}}(y) \setminus G^{i}_{\text{rnd}}(y)) = \frac{k_{i+1}}{|G|}\left(1 - \frac{k_i}{|G|}\right) \tag{24}$$

The probability of at least one violation among $|G|$ classes:

$$\Pr(E^c_i) = 1 - \left(1 - \frac{k_{i+1}}{|G|}\left(1 - \frac{k_i}{n}\right)\right)^{|G|} \tag{25}$$

In our experiments typical hierarchies are about $k_{i+1} = |G|/(2L)$ and $k_i = 2k_{i+1} = |G|/L$:

$$\frac{k_{i+1}}{|G|}\left(1 - \frac{k_i}{|G|}\right) = \frac{1}{2L}\left(1 - \frac{1}{L}\right) = \frac{L-1}{2L^2} \tag{26}$$

Thus:

$$\Pr(E_i) \geq 1 - \exp\left(-\frac{|G|(L-1)}{2L^2}\right) \tag{27}$$

By union bound:

$$\Pr\left(\bigcup_{i=1}^{L-1} E^c_i\right) \geq \max_i \Pr(E^c_i) = \Omega(1/L) \tag{28}$$

Therefore, random supervision leads to violations with constant probability, independent of training quality or sample size. The above probability multiplied by PAC bound gives the desired bound. □

Theorem 2.

Theorem (Explicit Minimisation of Ordering Mismatches for Attributes). For an input x with true label y, define:

$$\mathcal{A}_i(x) = \{a \mid \hat{a}^c_i(x) \geq \tau_a\}, \tag{29}$$

where $\tau_a \in [0,1]$ is a threshold, typically chosen as 0.5. Then, given $|\widehat{\ell_{\texttt{total}}} - \ell^*_{\texttt{total}}| \leq \epsilon$,

$$\Pr_{f^{\texttt{foca}}}\left(\mathcal{A}_i(x) \not\subseteq \mathcal{A}_{i+1}(x)\right) << \Pr_{f^{\texttt{rnd}}}\left(\mathcal{A}_i(x) \not\subseteq \mathcal{A}_{i+1}(x)\right). \tag{30}$$

Proof. The proof is similar to Theorem 1, except with the ordering changed. □

Proof of Theorem 4.2

Theorem 4.2 (Information-Theoretic Benefit of FCA Supervision)**.** *Consider a FoCA-CBM trained with formal concept lattice $\mathcal{L}$ constructed from $\langle G, M, I\rangle$. Let network layer j be supervised by lattice level i via class groups $G^{(i)}$ and attribute sets M_i. Then, under bounded training error $|\hat{\ell} - \ell^*| \leq \epsilon$ with N training samples, the ϵ-calibrated information gain of the network for layer j is:*

$$I_{\mathcal{D}}(f_j(X); Y) - I_{\mathcal{D}}(f_{j-1}(X); Y) \geq \Delta^{(i)}_{\text{lattice}} - 2\Delta_{\text{align}}(\epsilon),$$

where $\Delta_{\text{align}}(\epsilon) = O\big(\sqrt{\epsilon \log |G|} + N^{-1/2}\big)$.

Notation. For a sample (x, y) let

$$\hat{s}_j[c] := s_j[c] \cdot \sigma(s_{j-1}[c]) \qquad \text{and} \qquad p_j[c] := \sigma(\hat{s}_j[c])$$

where $\sigma(z) = 1/(1 + e^{-z})$. Let $m := |G^i(y)|$. The group BCE loss at layer j is

$$\ell^{\text{group}}_{\text{BCE},j}(x, y) = -\sum_{c \in G^{(i)}(y)} \log p_j[c] - \sum_{c \notin G^{(i)}(y)} \log(1 - p_j[c]).$$

Assumption A3.6 (BCE Loss Bound on Groups)**.** For layer j supervised by lattice level i, the group-level BCE loss satisfies:

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\ell_{\text{BCE}_j}^{\text{group}}\right] = \mathbb{E}_{(x,y)}\left[\sum_{c\in G}\mathbb{1}_{c\in G^{(i)}(y)}(-\log\sigma(\hat{s}_j[c])) + \mathbb{1}_{c\notin G^{(i)}(y)}(-\log(1-\sigma(\hat{s}_j[c])))\right] \le \beta\epsilon, \tag{31}$$

where $\beta > 0$ is the weight from Equation (3), and $\hat{s}_j = \sigma(s_j \cdot \sigma(s_{j-1}))$ is the refined prediction.

Assumption A3.7 (Cluster-Density Matching)**.** Layer j is selected via the class-cluster density procedure such that:

$$|\text{ClassClusterDensity}(j) - \mathbb{E}_{\text{fc}\in\mathcal{L}_i}[|\text{fc.extent}|]| \le \gamma, \tag{32}$$

where $\gamma = O(N^{-1/2})$ from finite-sample concentration.

Lemma 1 (Average sigmoid is close to 1 on the true group). For any sample (x, y) whose per-sample group loss satisfies $\ell_{\text{BCE},j}^{\text{group}}(x, y) \le K\epsilon$ (which holds for most samples by Markov/Chebyshev from the expectation bound), we have

$$\frac{1}{m}\sum_{c\in G^{(i)}(y)} p_j[c] \;\ge\; \exp\Big(-\frac{K\epsilon}{m}\Big) \;=\; 1 - O\Big(\frac{\epsilon}{m}\Big). \tag{33}$$

Proof. For the classes in the true group apply Jensen to $-\log$:

$$-\log\Big(\frac{1}{m}\sum_{c\in G^{(i)}(y)} p_j[c]\Big) \le \frac{1}{m}\sum_{c\in G^{(i)}(y)} -\log p_j[c] \le \frac{K\epsilon}{m}.$$

Exponentiating yields (33). □

Lemma 2 (Pigeonhole / concentration - most individual sigmoids are high). Define $a := \dfrac{K\epsilon}{m}$. For any $\delta \in (0, 1)$ let α be the fraction of classes $c \in G^i(y)$ with $p_j[c] \le 1 - \delta$. Then

$$\alpha \le \frac{a}{\delta}.$$

Choosing $\delta = \sqrt{a}$ gives $\alpha \le \sqrt{a} = O\Big(\sqrt{\frac{\epsilon}{m}}\Big)$. Hence at least a $1 - O(\sqrt{\epsilon/m})$ fraction of classes in the true group satisfy

$$p_j[c] \ge 1 - O\Big(\sqrt{\frac{\epsilon}{m}}\Big).$$

Proof. If αm classes have $p_j[c] \le 1 - \delta$ then the group average is $\le \alpha(1 - \delta) + (1 - \alpha) \cdot 1 = 1 - \alpha\delta$. Combine with (33) to get $\alpha\delta \le a$, hence $\alpha \le a/\delta$. □

Lemma 3 (From high sigmoid to large refined logit and per-class logit separation). Let $\tau := \sqrt{\dfrac{K\epsilon}{m}}$ (so $\tau \to 0$ with $\epsilon \to 0$). For the majority of classes in $G^i(y)$ (fraction $1 - O(\tau)$) we have

$$p_j[c] = \sigma(\hat{s}_j[c]) \ge 1 - \tau \quad\Longrightarrow\quad \hat{s}_j[c] \ge \sigma^{-1}(1 - \tau) = \log\frac{1-\tau}{\tau} =: L_j^+.$$

Since $\hat{s}_j[c] = s_j[c] \cdot \sigma(s_{j-1}[c])$ and (by earlier-level supervision) for classes that were kept at layer $j - 1$ we have $\sigma(s_{j-1}[c]) \ge 1/2$ (for well-trained earlier layer), we deduce for those classes

$$s_j[c] \ge \frac{L_j^+}{\sigma(s_{j-1}[c])} \ge L_j^+.$$

Similarly, for most classes outside the true group one obtains $p_j[c] \le \tau$ and hence $\hat{s}_j[c] \le \sigma^{-1}(\tau) = -\log\frac{1-\tau}{\tau} =: L_j^- < 0$ and therefore $s_j[c] \le L_j^-/\sigma(s_{j-1}[c]) \le L_j^- < 0$. Thus the majority of in-group logits are $\ge L_j^+$ and the majority of out-group logits are $\le L_j^-$, and

$$L_j^+ - L_j^- \;=\; \Theta(\log(1/\tau)) \;=\; \Theta\Big(\tfrac{1}{2}\log\frac{m}{\epsilon}\Big).$$

Step 4 (Conditional entropy alignment). Fix a typical sample (x, y) for which the sigmoid-concentration conclusions hold, and let ζ denote the total probability mass (under the model's per-class probabilities implied by the refined sigmoids) on classes outside the true group $G^i(y)$. From Lemmas 1–3 we have $\zeta = O(\tau) = O\big(\sqrt{\frac{\epsilon}{m}}\big)$ (up to constants and averaging effects). The conditional entropy of Y given the representation $f_j(x)$ satisfies

$$H(Y \mid f_j(x)) \le (1-\zeta)\log m + H_{\text{bin}}(\zeta) + \zeta\log(n),$$

where $H_{\text{bin}}(\zeta) = -\zeta\log\zeta - (1-\zeta)\log(1-\zeta) = O(\zeta\log(1/\zeta))$. Therefore

$$H(Y \mid f_j(x)) = \log m + O\big(\zeta\log n\big) = \log|G^{(i)}(y)| + O\Big(\sqrt{\frac{\epsilon}{m}}\log n\Big).$$

Taking expectation over (x, y) and combining with finite-sample error from the class-cluster density matching (Assumption $\gamma = O(N^{-1/2})$) yields

$$\Big|H_{\mathcal{D}}(Y \mid f_j(X)) - \mathbb{E}_y[\log|G^i(y)|]\Big| \le \Delta_{\text{align}}(\epsilon),$$

with

$$\Delta_{\text{align}}(\epsilon) = O\Big(\sqrt{\epsilon\log n} + N^{-1/2}\Big).$$

This is the claimed alignment error.

Final step (Information-gain comparison). Using $I(Y; f_j(X)) = H(Y) - H(Y \mid f_j(X))$ and the above alignment,

$$I_{\mathcal{D}}(f_j(X); Y) = H(Y) - \mathbb{E}_y[\log|G^i(y)|] \pm \Delta_{\text{align}}(\epsilon) = I^{(i)}_{\text{lattice}}(Y) \pm \Delta_{\text{align}}(\epsilon).$$

Similarly for layer $j-1$ aligned with level $i-1$ we get

$$I_{\mathcal{D}}(f_{j-1}(X); Y) = I^{(i-1)}_{\text{lattice}}(Y) \pm \Delta_{\text{align}}(\epsilon).$$

Subtracting,

$$I_{\mathcal{D}}(f_j(X); Y) - I_{\mathcal{D}}(f_{j-1}(X); Y) \ge \Delta^{(i)}_{\text{lattice}} - 2\Delta_{\text{align}}(\epsilon). \quad \square$$

A4. More Analysis

Impact of Class-Ordering: We empirically validate Theorem 4.1 by training a FoCA CBM on ImageNet100 with random class ordering. Here, a sample is supervised by the right group with a probability of 0.5 and is supervised by a wrong group with a probabililty of 0.5. We observe that this leads to a sharp fall in test accuracy (91.36 → 28.96), showing the importance of class ordering.

Alternative Hierarchies: The prompt used for GPT4 to generate the LLM-Based hierarchy was the following: *Given a set of classes and attributes, generate 2 sets: a set of general attributes and a set of specific attribute, with the set of specific attributes being a superset of general attributes. I also need a class group set that would get activated using the general attributes per class. For example: a general set of attributes could be "animal", "vertebrate", "mammal", "strong" and a specific set of attributes could be "animal", "vertebrate", "a long beak", "large wings", "mammal", "rocks", "strong" and for the class "macaw", a class group could be "indigo bunting, "macaw", "flamingo" which could be activated by a subset of the general attributes.*

This generates a two-level hierarchy which is compared with a two-level FoCA CBM.

Choice of α and β: We perform ablation studies on the two hyperparameters α and β. We fix one of the values of these hyperparameters from the best models and vary the other one. Both are varied among [0.01, 0.1, 1]. The results are reported

Table A9. Effect of α on FoCA CBM on accuracy.

Dataset	Alpha	FoCA CBM
AWA2	0.01	92.36
	0.1	92.13
	1	91.94
CIFAR100	0.01	79.36
	0.1	75.84
	1	73.50
Imagenet100	0.01	91.92
	0.1	87.73
	1	87.56

Table A10. Effect of β on FoCA CBM on accuracy.

Dataset	Beta	FoCA CBM
AWA2	0.01	92.36
	0.1	91.24
	1	90.02
CIFAR100	0.01	79.36
	0.1	71.85
	1	65.09
Imagenet100	0.01	91.92
	0.1	86.94
	1	85.22

in Tables A9 and A10. A lower value is preferred since the first two terms of our loss function in Eqn 4 are summed up over multiple layers, thus scaling the loss values.

More Cluster-based Analysis: We provide more comprehensive results on our cluster-based analysis here. Going beyond the reported average *CI* and *DBI* scores in Table 1, we examine herein how these metrics evolve across blocks of a model. Ideally, both scores should decrease as we move deeper into a network, indicating lower cluster impurity and more compact, well-separated representations. In other words, we would want embeddings to move closer to the origin of the CI-DBI space across blocks. As is evident from Figure A6 for CIFAR-100, the FoCA CBM models move closest to the origin. In contrast, some baselines show little change across layers, suggesting weaker representation refinement. The *CI-DBI* plots for the other datasets are also provided. We also use line plots to visualize cluster impurity reduction alone through blocks (Fig A9) and see that our models gradually reduce in cluster impurity, unlike the other models where there is either a sharp drop in the last block or no drop at all. This indicates our models' ability to learn more meaningful embeddings. Finally, we provide more comprehensive t-SNE plots comparing the cluster formation in all blocks of the respective backbone networks in Fig A12, A13, A14. The set of classes considered for AwA2 were *dalmatian, german shepherd, horse*, for ImageNet100 were *agama, banded-gecko, whiptail* and for CIFAR100 were *beetle, butterfly, cockroach*.

A5. Dataset Details

- **AwA2:** The Animals with Attributes (AwA2) dataset (Xian et al., 2019) is commonly used for zero-shot learning (ZSL) and attribute-based classification. It consists of 37322 images (26125 training, 11197 testing) of 50 animal classes, annotated with 85 numeric attribute values for each class and is class-level expert annotated. Each class in the dataset has on average 31 ± 4 active attributes, with 22 being the minimum number of attributes active for a particular class and 39 being maximum.
- **CIFAR100:** CIFAR100 is a well-known subset of the Tiny Images dataset (Krizhevsky et al., 2009) comprising 100 classes. Attributes associated with each of these classes are generated using GPT-3 (Oikarinen et al., 2023), where an initial concept set is generated through queries like "List the most important features for recognizing something as a {class}". This is followed by a sequence of filtering steps, to improve the quality of the concept set. This involves the k-means clustering (with $k = 700$) of similar attributes together based on their `all-MiniLM-L6-v2` embeddings and choosing the closest attributes to each cluster centroid as the cluster representatives. The dataset finally consists of 60000 images (50000 for training and 10000 for testing), 700 attributes and 100 classes. Each class in the dataset has on average 16 ± 3 active attributes, with 8 being the minimum number of attributes active for a particular class and 24 being maximum.
- **ImageNet100:** ImageNet100 is a well-known subset of the ImageNet-1k dataset (Russakovsky et al., 2015). We randomly select 100 classes from the set of 1k classes. We follow the same process we followed for the CIFAR100 dataset here as well and acquire attributes from GPT-3. The dataset finally consists of 134973 images (129973 training, 5000 testing) of 100 distinct classes and 700 LLM-generated unique attributes. Each class in the dataset has on average 18 ± 3 active attributes, with 9 being the minimum number of attributes active for a certain class and 25 being maximum.

Some example classes and their corresponding attributes are provided in Tab A12.

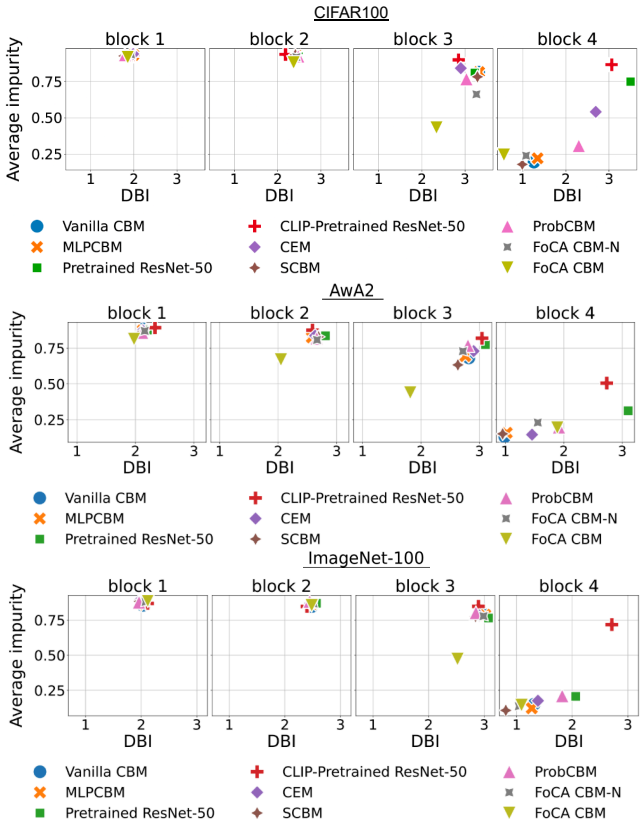


Figure A6. CI vs DBI plot per block on CIFAR100, AwA2 and ImageNet100 datasets. Each marker indicates a model trained on the respective dataset. Markers should ideally move towards the origin in higher blocks. We attribute the superiority of CBMs and MLPCBMs on AwA2 to the simplicity of the dataset.

Table A11. Details about the formal concept lattice obtained for each dataset.

Dataset	$\lvert G\rvert$	$\lvert M\rvert$	**Fill ratio**	$\lvert\mathcal{L}\rvert$	L	**Worst case** $\lvert\mathcal{L}\rvert$	**Time** (s)	**Space** (MB)
AwA2	50	85	0.368	64315	26	1.13×10^{15}	37.37	63.62
CIFAR100	100	700	0.023	915	10	1.27×10^{30}	1.97	1.00
ImageNet-100	100	700	0.026	1593	10	1.27×10^{30}	4.56	1.68

A6. Lattice Details

Our formal concept lattices are constructed using the `concepts` [1] Python module. This module employs the Fast Concept Analysis algorithm (Troy et al., 2007) for generating the lattices (ConstructLattice in Algorithm 1). The hierarchy level of each formal concept is computed by first performing a topological sort on the lattice (which is always a directed acyclic graph), and then iteratively updating the level of the upper neighbors of each formal concept traversed in topological order (described in Algorithm 2).

- The **AwA2** lattice consists of 50 classes, 85 attributes, and 64315 total formal concepts across 26 hierarchy levels with 1, 50, 743, 3038, 5755, 7440, 7876, 7472, 6680, 5738, 4800, 3912, 3083, 2310, 1693, 1221, 873, 613, 409, 262, 165, 98, 52, 23, 7, 1 formal concepts per level. Computing the lattice took 37.37 seconds on average, and it occupies 63.62 MB of space.
- The **CIFAR100** lattice consists of 100 classes, 700 attributes and 915 total formal concepts across 10 hierarchy levels with 1, 100, 345, 211, 131, 75, 33, 14, 4, 1 formal concepts per level. Computing the lattice took 1.97 seconds on

[1] https://pypi.org/project/concepts/

Algorithm 2 COMPUTE HIERARCHY LEVELS($\mathcal{L}$):

Require: Formal concept lattice $\mathcal{L}$.
  $\mathcal{L}_s \leftarrow \text{TopologicalSort}(\mathcal{L})$ ▷ Infimum at the first index
  level[fc] $\leftarrow 0 \quad \forall$ fc $\in \mathcal{L}_s$
  for u in $\mathcal{L}_s \setminus \{\mathcal{L}_s.\text{infimum}\}$ **do**
    for v in u.upper_neighbors **do**
      $\text{level}[v] = \max\{\text{level}[v], \text{level}[u] + 1\}$
    end for
  end for
  return level

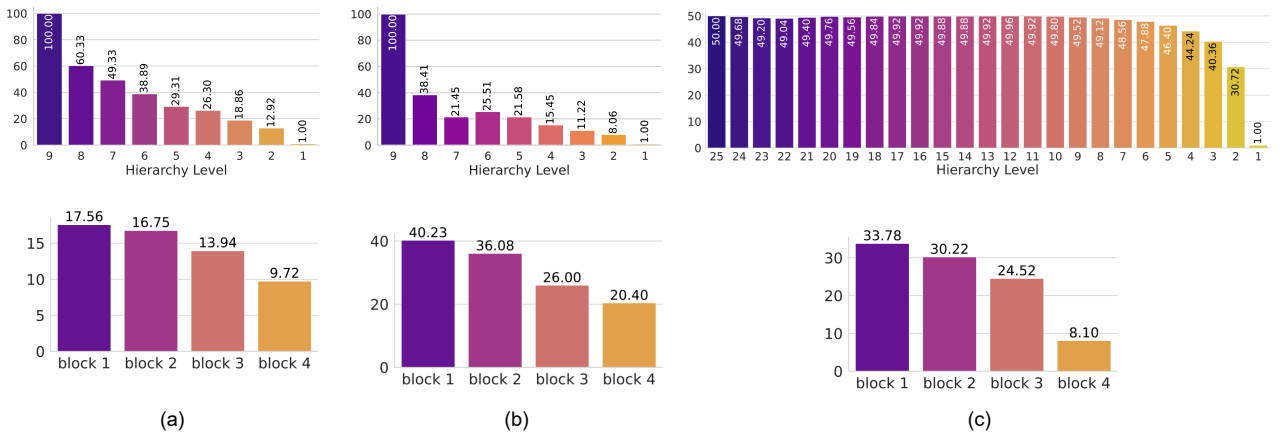


Figure A7. The average number of classes active per lattice level (top) and per block of a ResNet (bottom) for the ImageNet-100 (a), CIFAR100 (b) and AwA2 (c) datasets.

average, and it occupies 1.00 MB of space.

- The **ImageNet-100** lattice consists of 100 classes, 700 attributes, and 1593 total formal concepts across 10 hierarchy levels. The number of formal concepts in the 10 levels is 1, 100, 592, 415, 250, 129, 65, 30, 10, 1, respectively, going from the infimum to the supremum. Computing the lattice took 4.56 seconds on average, and it occupies 1.68 MB of space.

A7. Implementation Details

The algorithm for the whole method is provided in Algorithm 1.

Evaluation Metric Details: We provide additional details herein on the clustering-based metrics (*CI, DBI*) we report in our results in the main paper (Table 1). At each block, we get the set of embeddings on the samples for all n classes and cluster them (with $k = n$). For each cluster in a block, we compute the `gini-index` and `davies_bouldin_score`. Averaging this value over all the clusters in a block, gives the average impurity and average cluster compactness of that block. We then take the harmonic mean of this number across all blocks to get the numbers representing the average *CI* and *DBI* of the model. These are described in Algorithm 3 and 4.

Model Details: All models were run on a single NVIDIA GeForce RTX 3090. We use a ResNet18-based backbone for all AwA2 models and a ResNet50-based backbone for all CIFAR100 and ImageNet100 models. On our models, we place semantic layers at the end of blocks and hence have 4 backbone position choices corresponding to the 4 blocks in a ResNet. All the results reported in the main table are models with 2 semantic layers.

Algorithm 3 COMPUTE CLUSTER IMPURITY(f, $\mathcal{D}$):

Require: Model f, Dataset $\mathcal{D}$.
  Initialize ci = 0
  for b in f.blocks **do**
    for j in $\mathcal{D}$ **do**
      $E_{b_i} = E_{b_i} \cup \{h_i^j\}$ ▷ Set of embeddings at block b_i
    end for
    clusters = `k-means`(E_{b_i}, n) ▷ Cluster E_{b_i} with n centers
    for c in clusters **do**
      ci += `gini-index`(c)
    end for
    ci /= n
  end for
  ci /= #blocks
  return ci

Algorithm 4 COMPUTE DBI(f, $\mathcal{D}$):

Require: Model f, Dataset $\mathcal{D}$.
  Initialize dbi = 0
  for b in f.blocks **do**
    for j in $\mathcal{D}$ **do**
      $E_{b_i} = E_{b_i} \cup \{h_i^j\}$ ▷ Set of embeddings at block b_i
    end for
    clusters = `k-means`(E_{b_i}, n) ▷ Cluster E_{b_i} with n centers
    dbi += `davies_bouldin_score`(E_{b_i}, clusters.labels)
  end for
  dbi /= #blocks
  return dbi

Algorithm 5 CLASSCLUSTERDENSITYMODEL(f, $\mathcal{D}$):

Require: Model f, Dataset $\mathcal{D}$.
  Initialize avg_class_per_block $\leftarrow$ **0**
  $n \leftarrow$ number of classes in $\mathcal{D}$
  for b in f.blocks **do**
    for j in $\mathcal{D}$ **do**
      $E_{b_i} = E_{b_i} \cup \{h_i^j\}$ ▷ Set of embeddings at block b_i
    end for
    clusters = `k-means`(E_{b_i}, n) ▷ Cluster E_{b_i} with n centers
    for c in clusters **do**
      avg_class_per_block[b] += `UniqueClasses`(c) ▷ gets number of unique classes, associated with the embeddings, in a cluster c
    end for
    avg_class_per_block[b] /= n
  end for
  return avg_class_per_block

Algorithm 6 CLASSCLUSTERDENSITYLATTICE($\mathcal{L}$):

Require: Lattice $\mathcal{L}$
  Initialize avg_class_per_level $\leftarrow$ **0**
  for level in $\mathcal{L}$ **do**
    count $\leftarrow$ 0
    for `fc` in level **do**
      count += 1
      avg_class_per_level[level] += `len(fc.extent)`
    end for
    avg_class_per_level[level] /= count
  end for
  return avg_class_per_level

Algorithm 7 SELECTLAYERSANDLEVELS(f, $\mathcal{D}$, $\mathcal{L}$, m)

Require: Model f, Dataset $\mathcal{D}$, Lattice $\mathcal{L}$, number of (layer,level) pairs to select $m \geq 1$

```
block_density ← CLASSCLUSTERDENSITYMODEL(f, D)                    ▷ length = #blocks = L_f
level_density ← CLASSCLUSTERDENSITYLATTICE(L)                     ▷ length = #levels = L_L
L_f ← len(block_density)    L_L ← len(level_density)
selected_layers ← [ L_f − 1 ]                                     ▷ index of last (top) block
selected_levels ← [ L_L − 1 ]                                     ▷ index of most fine-grained lattice level
remaining ← m − 1
last_level_idx ← L_L − 1                                          ▷ we will search levels strictly above this index
for layer_idx in range(L_f − 2, −1, −1) do                        ▷ iterate remaining model blocks bottom→top
    if remaining == 0 then
        break
    end if
    layer_density ← block_density[layer_idx]
    chosen_level ← None                                           ▷ search lattice levels from (last_level_idx - 1) downward to 0
    for level_idx in range(last_level_idx - 1, -1, -1) do
        if level_density[level_idx] ≥ layer_density then
            chosen_level ← level_idx
            break
        end if
    end for
    if chosen_level is not None then
        selected_layers.append(layer_idx)
        selected_levels.append(chosen_level)
        last_level_idx ← chosen_level
        remaining ← remaining − 1
    end if
end for
return selected_layers, selected_levels
```

Computational Complexity: Our FoCA CBM models were trained in ≈: 40 min for AwA2, 1.75 hrs for CIFAR100 and 3 hrs for ImageNet100 on a single NVIDIA GeForce RTX 3090; these times were about the same timings as Vanilla CBMs and MLPCBMs took. Our lattices are generated offline before training and took ≈: 37 secs for AwA2, 2 secs for CIFAR100 and 4.5 secs for ImageNet100, thus making this a near-negligible cost.

Hyperparameter Details:

- *Vanilla CBMs and MLPCBMs*: All models here were trained for 30 epochs with a batch size of 128, an AdamW optimizer and a onecycle scheduler. The AwA2 and CIFAR100 models were trained using a learning rate of 3×10^{-4}, while the ImageNet100 models were trained with a learning rate of 1×10^{-4}.
- *Posthoc CBMs*: Here, we use the multimodal CLIP-based backbones for AwA2 and ImageNet100 datasets, with $\lambda = 2 \times 10^{-4}$ and a batch size of 512. For CIFAR100, we take the numbers from the respective paper (Posthoc CBMs).
- *Label-Free CBMs*: Most hyperparameters used here were the same as the ones reported by the paper on the ImageNet dataset. Additionally, we use a clip-cutoff of 0.26 for ImageNet100 models and 0.25 for AwA2 models. An Adam optimizer with a learning rate of 1×10^{-3} was used to learn the concepts. Finally, for learning the classes, we use `glmsaga` with a regularization strength of 1×10^{-4} for ImageNet100 and 3×10^{-4} for AwA2. We use a batch size of 512. We report the CIFAR100 results from the paper.
- *Concept Embedding Models*: We train all these models for 100 epochs with a batch size of 256, a learning rate of 0.01 and an SGD optimizer.
- *Language in a Bottle (LaBo)*: Since these models work with CLIP-based backbones, we use a CLIP:RN50, along with submodular concept selection, max epochs of 10000, batch size of 512 and learning rate of 1×10^{-5} on all datasets.
- *Stochastic CBMs*: All models were trained for 70 epochs, with a batch size of 64, learning rate of 3×10^{-5} using an Adam optimizer, with the number of monte carlo samples being 100.

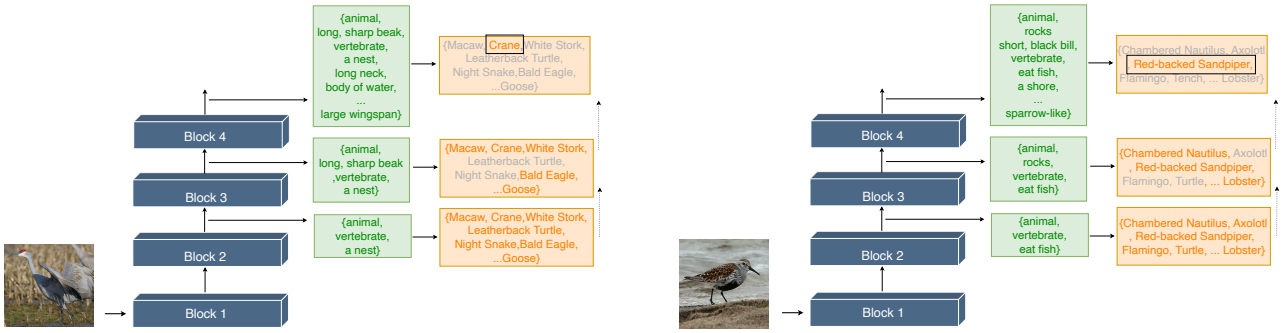


Figure A8. Some qualitative results of the predicted attributes and subsequent class group refinement for some samples from ImageNet100.

- *Probabilistic CBMs*: All models were trained with an embedding size of 16, training intervention probability of 0.25, learning rate of 1×10^{-2} with an SGD optimizer, a batch size of 256 and max epochs of 100.
- *Coarse-to-Fine CBMs*: For all the models here we first compute the CLIP similarities using a CLIP:RN50 model and then train the models. We use a learning rate of 3×10^{-4} with an Adam optimizer, batch size of 256 and number of epochs of 30.
- *Hybrid CBMs*: Since these models work with CLIP-based backbones, we use a CLIP:RN50 on all datasets, along with submodular concept selection with a dynamic concept ratio of 0.5, max epochs of 5000 and learning rate of 5×10^{-5}. For the AwA2 and CIFAR100 models, we use a batch size of 512 and for the Imagenet100 models, we use a batch size of 4096.
- *FoCA CBMs*: Our hyperparameters here are the same as Vanilla CBM models.

A8. Limitations

To facilitate future work, we also outline a few limitations of our approach. Firstly, as with all concept-based methods, the quality of our intermediate semantic representations is dependent on the accuracy of the attribute annotations. This dependency is particularly pronounced in LLM-annotated datasets such as CIFAR100 and ImageNet100. Enhancing annotation quality could therefore lead to notable gains in both model performance and interpretability. Secondly, once again mirroring a concern that is common across concept-based models, constraining the model to operate through semantic concepts can, in some cases, limit overall performance, a trade-off that is reflected in parts of our results. While we demonstrate consistent improvements in interpretability and related metrics, narrowing this performance gap remains a key area for further investigation. Finally, concept-based models typically treat concepts as static entities. Developing mechanisms that allow concept representations to adapt dynamically to the context of a specific input (, a given input image) could be an interesting approach to improvements in flexibility and model performance.

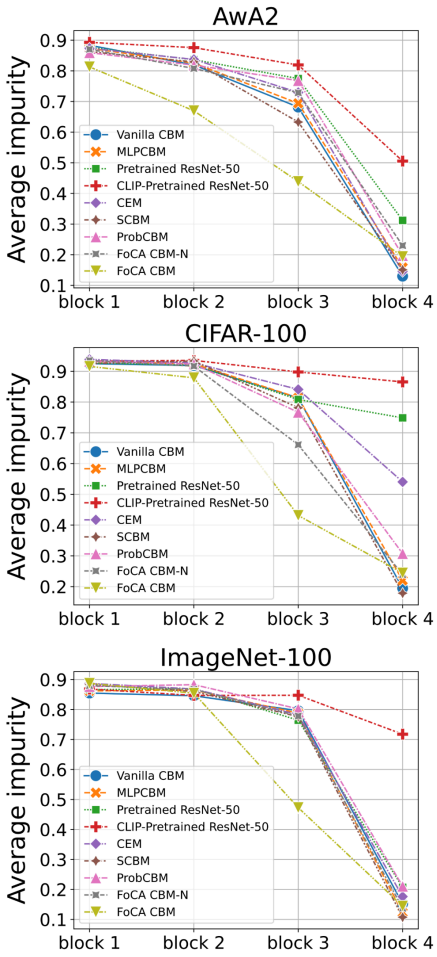


Figure A9. Cluster impurity over all models per block on AwA2, CIFAR100 and ImageNet100 datasets. Our models (inverted light green triangle) display a gradual reduction in impurity. The other models fall sharply at the last block or not at all.

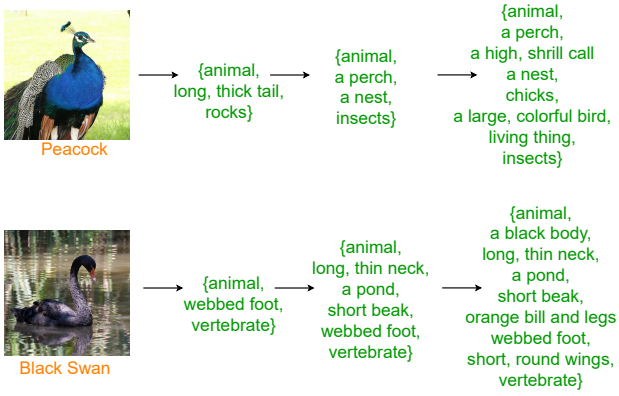


Figure A10. More examples of the attributes learned by a FoCA CBM after blocks 2, 3 and 4 of a ResNet50 for the classes *Peacock* and *Black Swan* in ImageNet100.

Table A12. Examples of classes and subsets of corresponding attributes from AWA2, CIFAR100 and ImageNet100 datasets.

Dataset	Class	Concepts
AwA2	Raccoon	black, white, gray, patches, spots, stripes, furry, small, pads, paws, tail, chewteeth, meatteeth, claws, walks, fast, quadrapedal, active, nocturnal, hibernate, agility
	Cow	black, white, brown, patches, spots, furry, toughskin, big, bulbous, hooves, tail, chewteeth, horns, smelly, walks, slow, strong, quadrapedal, active, inactive
	Dolphin	white, blue, gray, hairless, toughskin, big, lean, flippers, tail, chewteeth, swims, fast, strong, muscle, active, agility, fish, newworld, oldworld, coastal, ocean, water
CIFAR100	Chair	furniture, a person, object, legs to support the seat, an office, a computer, a desk, four legs, a backrest, armrests on either side
	House	windows, building, object, structure, a yard, a chimney, a door, a wall, siding or brick exterior, a garage, roof
	Kangaroo	a grassland, short front legs, an animal, a safari, mammal, a long, powerful tail, brown or gray fur, marsupial, long, powerful hind legs, Australia
ImageNet100	Electric Ray	paddle-like fins, a flat circular shape, fish, a long, thick tail, mammal, animal, water, vertebrate, a large mouth, a large, bulky body
	White Stork	an animal, a large size, a tree, a field, insects, a sky, a long, curved neck, white feather, a thin neck, long red legs, long, arms and legs, a medium-sized body, vertebrate, a long orange beak
	Komodo Dragon	a large size, a keeper, scales, a tree, a dish, scaly skin, a rock, long, sharp claws, a long, thick tail, a long, forked tongue, an animal, reptile, a fence, vertebrate, a water dish, a zoo, a heat lamp, a large, bulky body, a cage, a lizard

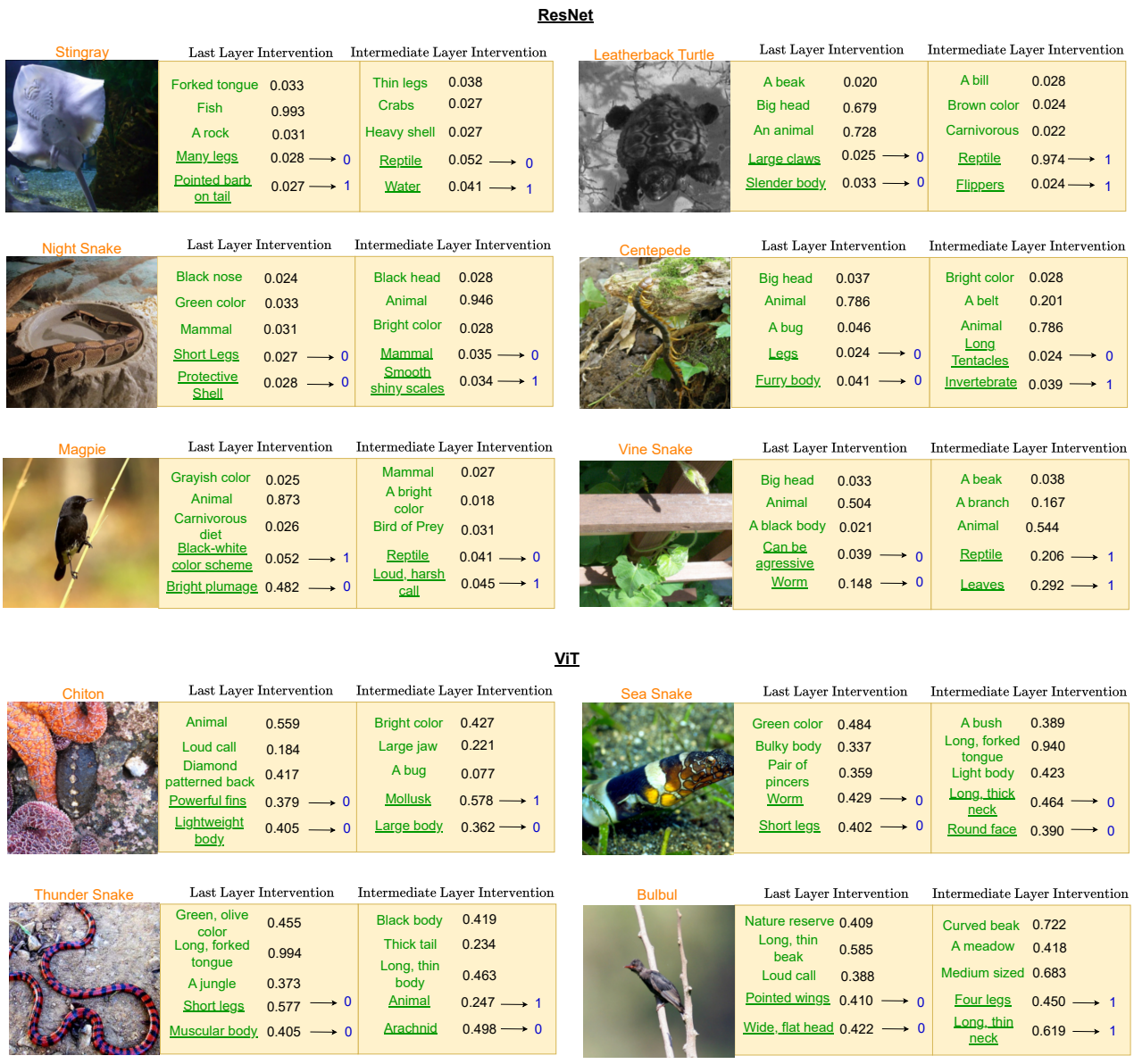


Figure A11. More examples of the kind of attributes intervened on in the last layers versus an intermediate layer (chosen according to the severity of the misclassification). Intermediate layers have more general attributes.

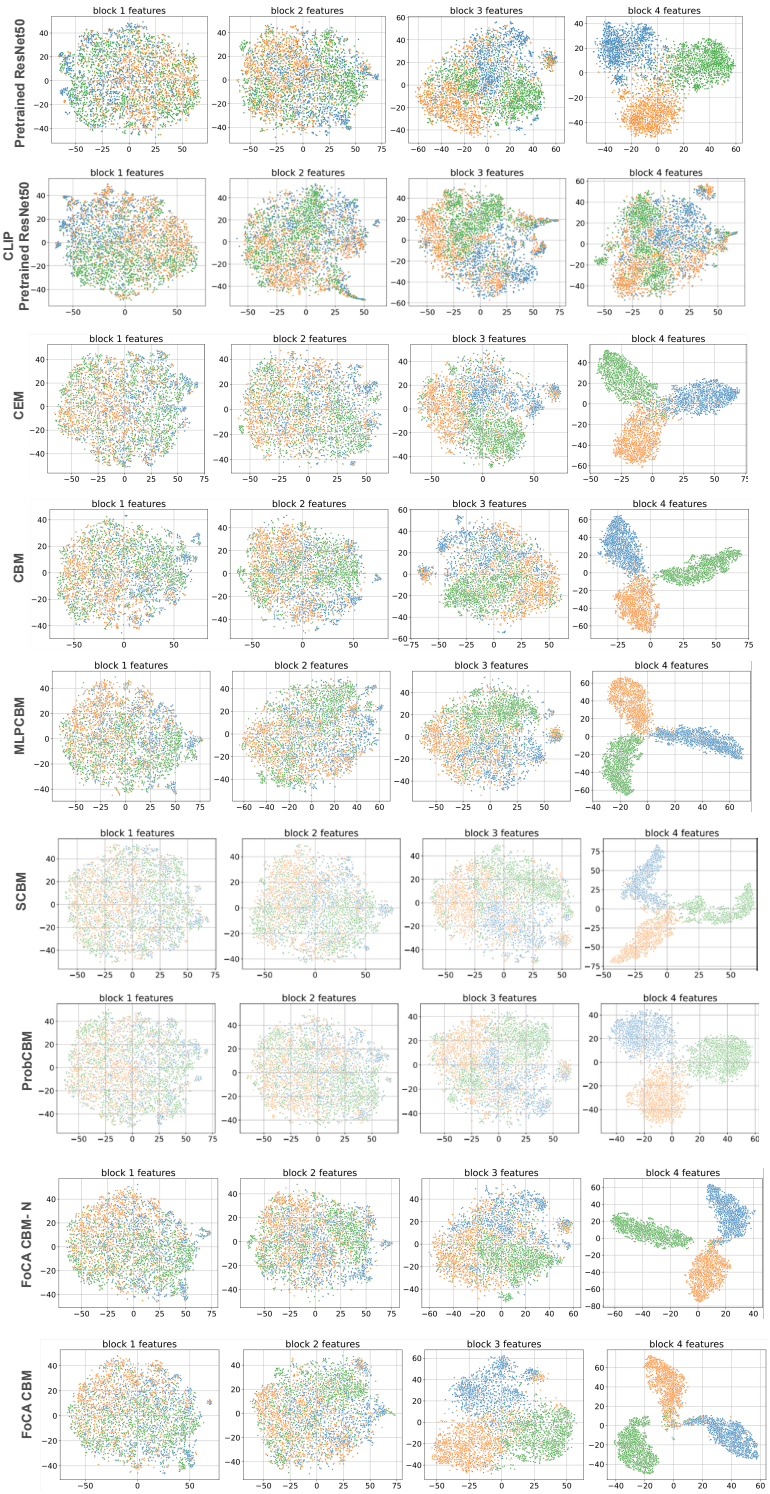


Figure A12. t-SNE plots of the embeddings obtained from the backbones of the models mentioned on the left of each plot on ImageNet100. On most models the clusters separate out only at the final block; in FoCA CBM, it happens gradually over blocks.

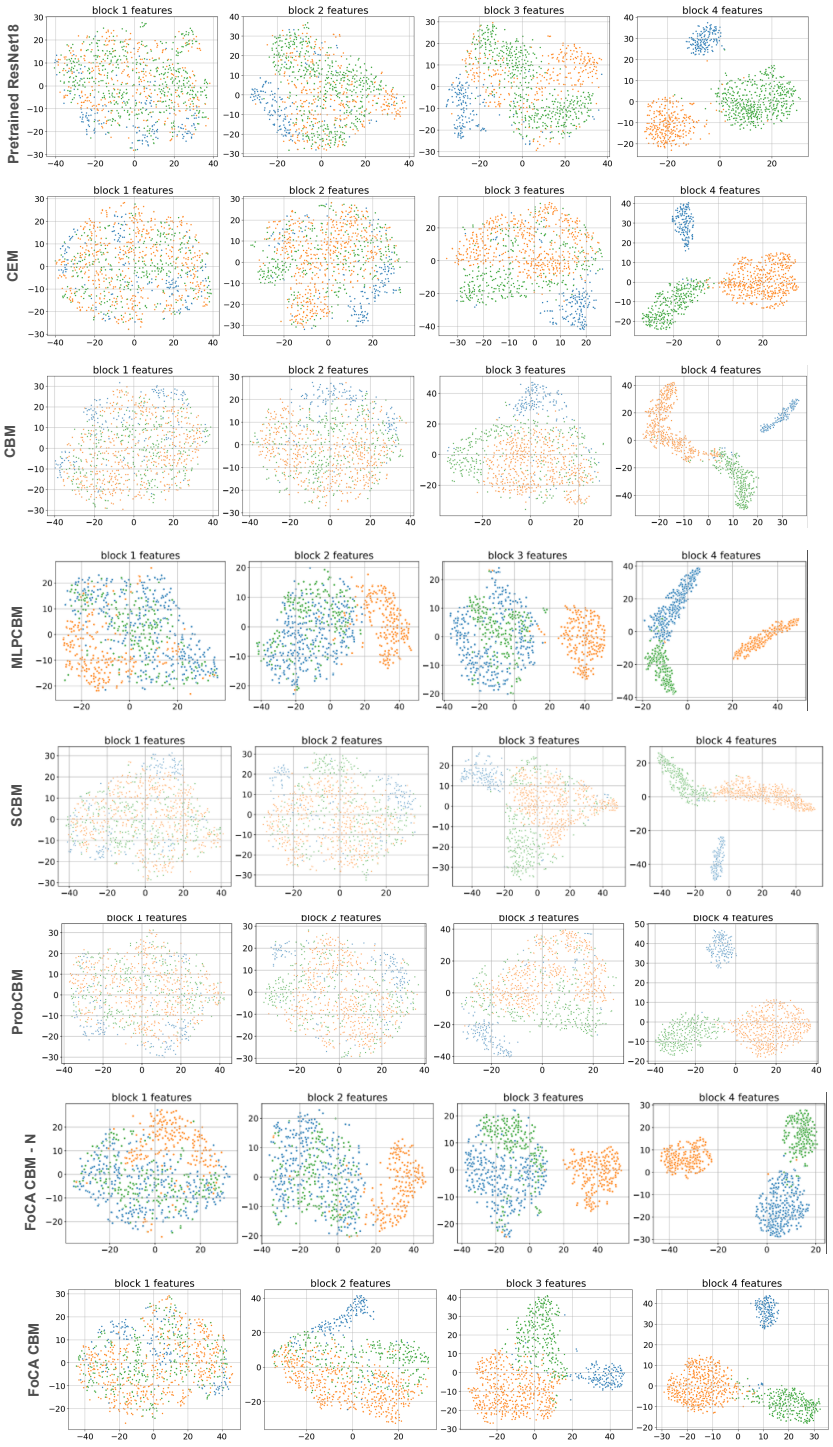


Figure A13. t-SNE plots of the embeddings obtained from the backbones of the models mentioned on the left of each plot on AwA2. On most models the clusters separate out only at the final block; in FoCA CBM, it happens gradually over blocks.

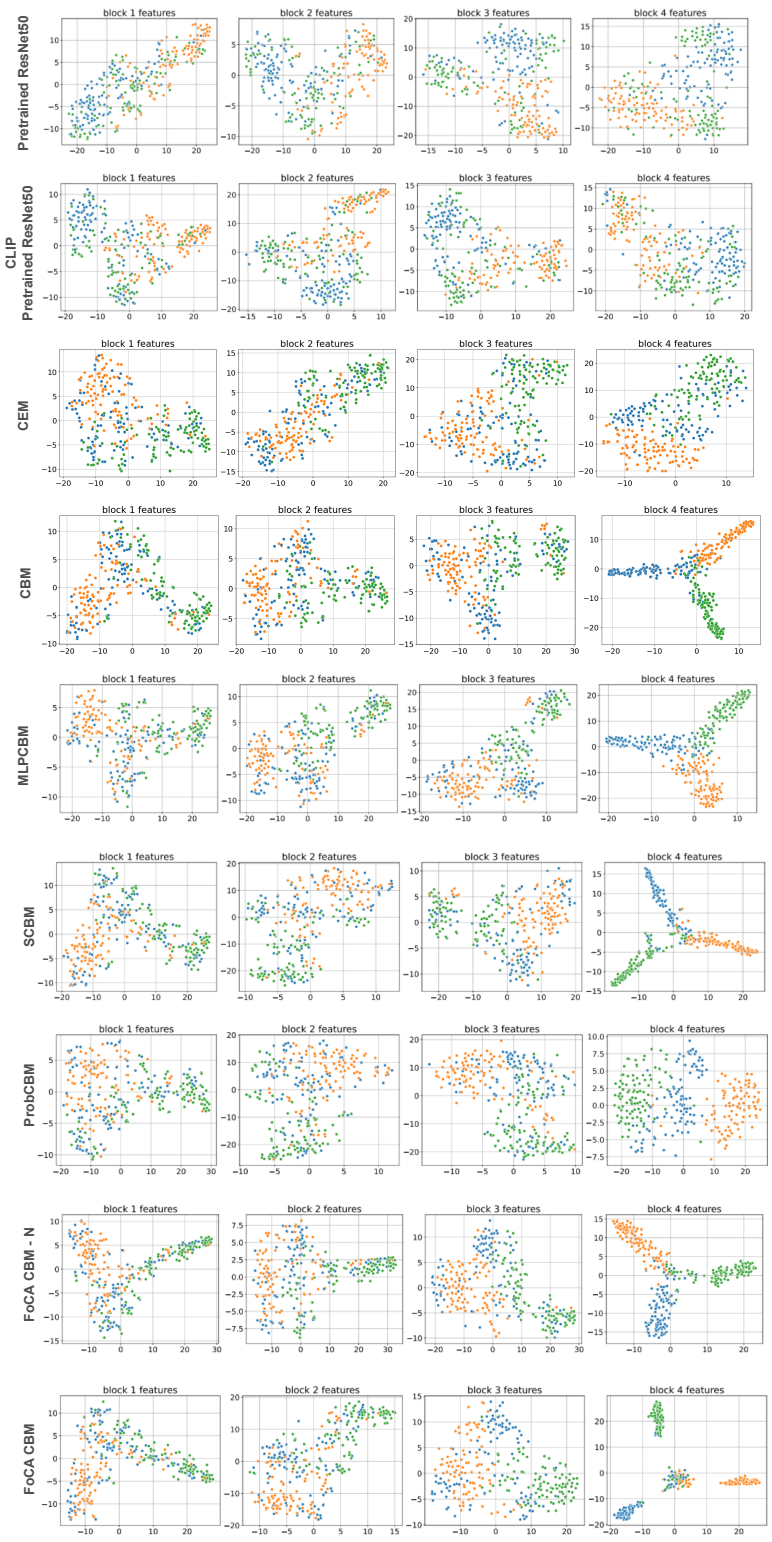


Figure A14. t-SNE plots of the embeddings obtained from the backbones of the models mentioned on the left of each plot on CIFAR100.

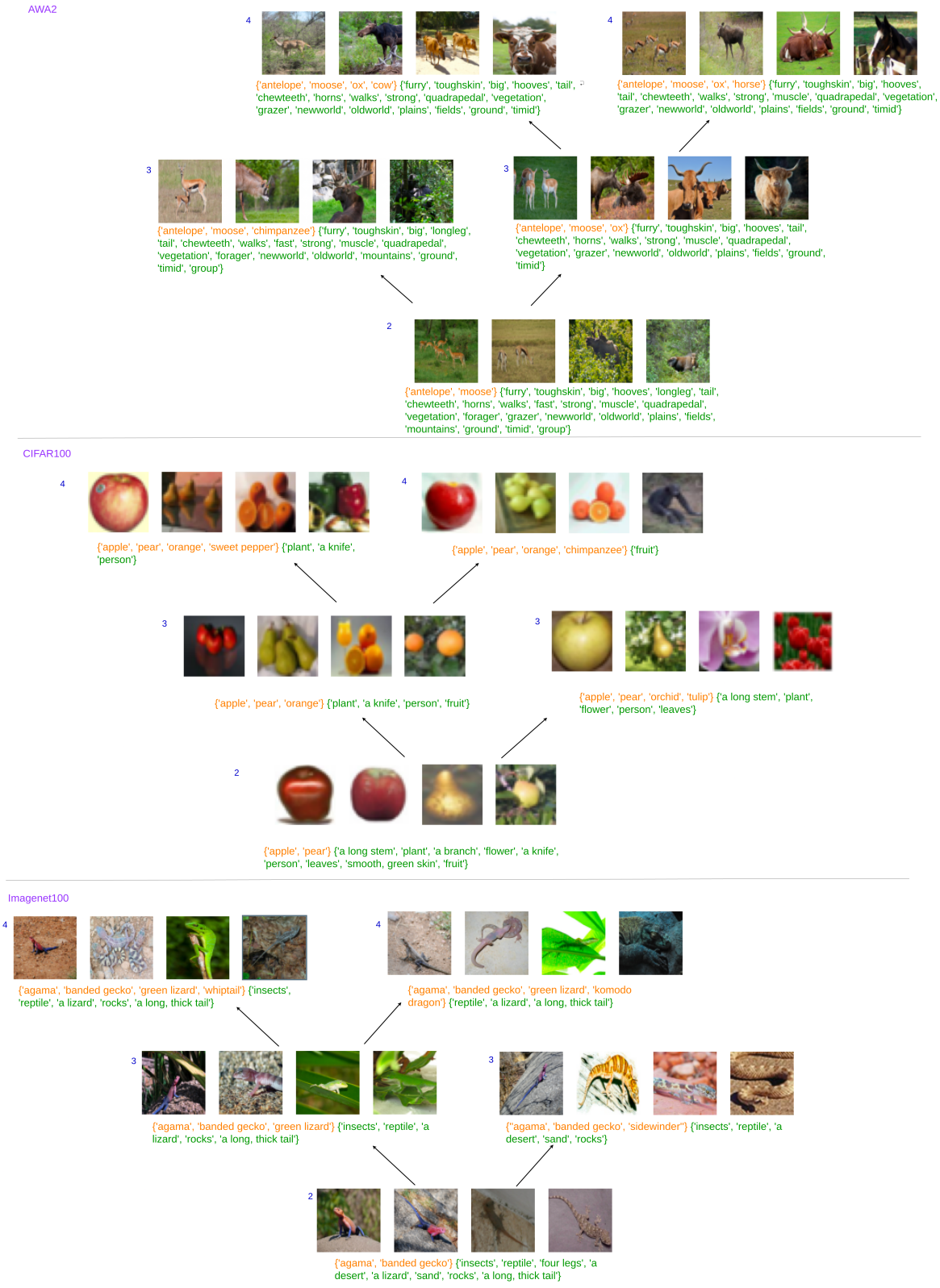


Figure A15. More examples of formal concepts (extent (classes) - intent (attributes)) from the lattices built for the AwA2, CIFAR100 and ImageNet100 datasets. The shown formal concepts have parent-children relations denoted by the arrows. The level that the formal concept belongs to is provided at the top left of each formal concept.