Learning Hierarchically using Formal Concepts

Deepika Vemuri
IIT Hyderabad
ai22resch11001@iith.ac.in

Sayanta Adhikari
IIT Hyderabad
ai22mtech12005@iith.ac.in

Ankit Saha
IIT Hyderabad
ai21btech11004@iith.ac.in

Vineeth N Balasubramanian
IIT Hyderabad
vineethnb@cse.iith.ac.in

Abstract

Learning semantics is crucial for deep learning models to be trustworthy and more aligned with human-like reasoning. Concept-based models offer a promising approach by learning classes in terms of interpretable semantic abstractions. However, two key limitations in such an approach are: 1. concepts of varying degrees of granularity are all learned in the same layer, with the same number of parameters, 2. as the concept layer comes right before the classifier, the network that the concepts are learned from still largely remains a black-box. In order to address these challenges, we propose a method for distributing concepts across the network. We use Formal Concept Analysis (FCA) to build a hierarchy that informs where in the network specific concepts are learned based on their level of abstraction. Our experiments on real-world datasets demonstrate the effectiveness of our approach by introducing a way to obtain staged semantically grounded representations.

1. Introduction

Humans learn conceptually [28]. This paradigm of learning has gained prominence lately from an interpretability perspective where concepts are introduced as neurons in the penultimate layer of a network [11]. These concept-based models learn in a two-step process - images are used to learn concepts and the concepts are then used to learn classes - and offer a certain level of interpretability because the classes are predicted in terms of the concepts. Now, humans not only learn conceptually but are also known to organize the concepts they acquire hierarchically [1, 8]. Current concept-based models, on the other hand, are still largely data-driven and lack this kind of structural organization. Models that mirror this sort of learning would naturally be more interpretable and more aligned with human-like reasoning.

It is empirically known that deep learning models exhibit hierarchical behavior, i.e. the earlier layers capture more abstract properties like texture and the deeper ones more class-specific properties [27]. However, the way that concept-based models are implemented doesn't align with this behavior, where concepts of varying degrees of granularity are all placed in the same layer and have the same number of parameters. We argue that it is more cognitively plausible for concepts to be spread throughout the network according to their level of abstraction. The key question is: *How do we train a concept-based neural network that can learn concepts of different granularities at different layers?*

We are inspired by the mathematical theory of Formal Concept Analysis (FCA [4]) and build a formal concept lattice using classes and text-based concepts (which we henceforth refer to as attributes). Our hypothesis is that with the abstract attributes a model is able to learn in the earlier layers in the network, it may not be able to discriminate between all classes with high confidence but it will be able to predict a certain group of classes (an extent). For example, given that the model has identified that an image contains the attributes: *fur, whiskers* and *claws*, it could say that the image might be either a *dog, cat* or a *racoon* with some certainty. The more specific the attributes get, the more the model should be able to refine its predictions, i.e. reduce the class group size.

We introduce a method to guide the learning process using the grouping information derived from a formal concept lattice. More specifically, we propose inserting attribute-classification layer pairs at intermediate positions within a given backbone architecture, using the hierarchical information derived from the formal concept lattice to determine which set of attributes to learn at each layer based on their level of abstraction. A model with such a capability could give us looking glasses into the internals of the model and as such would give us access to intermediate semantics.

Our key contributions are summarized below:

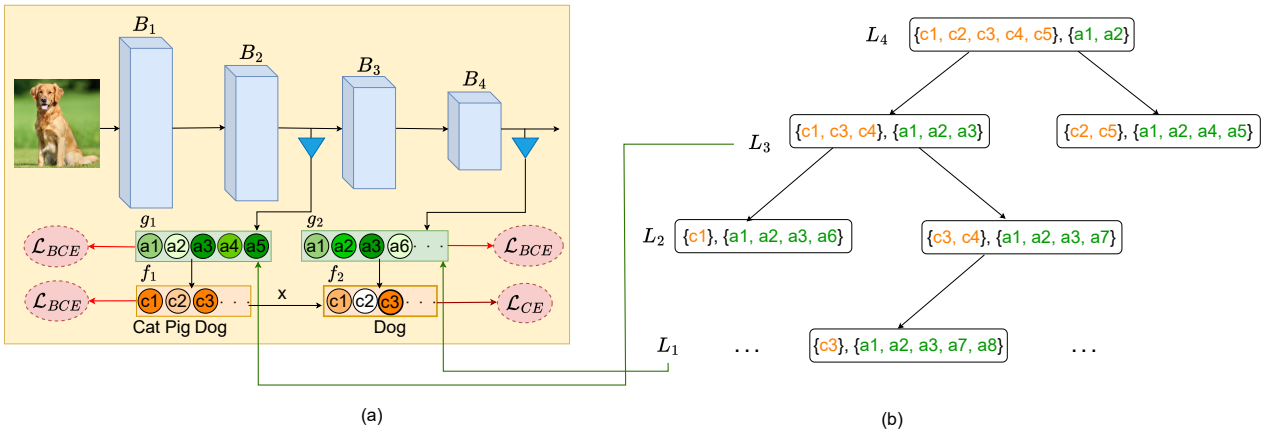

Figure 1. **Our Architecture:** (b) We construct a formal concept lattice using class-level attribute annotations. (a) The attribute and class grouping information at selected levels in the lattice is used to supervise learning at selected positions in the network (green arrows). Blue triangles indicate global average pooling layers (GAP), green layers are attribute layers and orange layers are classifiers. A darker shade in the circles indicates a higher strength.

- We extend ideas from FCA to a concept-based learning setting and propose a method to extract hierarchical supervisory information from a formal concept lattice.
- We present a new viewpoint to formalize the notion of intermediate semantics in concept-based models. Our method is general and can be used to induce intermediate semantics in arbitrary backbones.
- We experimentally validate our method on real-world datasets and investigate the impact of the chosen backbone positions and hierarchy levels.

2. Related Work

Concept-Based Models. Building inherently interpretable models using concepts is an actively growing area of research, originally introduced by [11]. There are several works that improve various aspects of these models like addressing concept leakage [14], including uncertainity quantification [10] and improving robustness [22]. Other efforts include increasing the model capacity using additional unsupervised concepts [21] and building concept bases for such models [26]. In all of these efforts, the network that learns the concepts still largely remains a black-box, which is an aspect we focus on in this work.

Hierarchical Learning. There are several aspects to hierarchical learning. One line of work learns hierarchical embeddings like order embeddings [24], hyperbolic entailment cones [3] and Poincaré embeddings [15]. Another line of work uses a hierarchy to augment the learning process. This is usually done by constraining the predictions of the model to obey the hierarchy [5, 6, 12]. We on the other hand, derive supervisory information from a hierarchy. Most CBM-based works that use hierarchies are limited to two-level ones [17, 23]. Our formal concept hierarchies can have any number of levels (26 levels on one of the datasets we use).

Formal Concept Analysis. This is a mathematical theory of data analysis where a set of objects and attributes are used to derive a formal concept hierarchy. There have been efforts to use this theory to learn more meaningful embeddings in deep learning settings: to encode closure operators in a neural network [19], to introduce an embedding technique [2] for problems with formal context-like structures like bipartite graphs [18] and for order-based representations using binary vectors inspired from FCA [7]. 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 induce intermediate semantics.

3. Methodology

Preliminaries and Notation:

Concept-Based Models [10, 11, 16]: We follow the setup introduced by [11] and define a concept-based model as a model that learns a mapping from $X \mapsto Y$ via an intermediate concept encoder $g(\cdot)$. These models learn from a three-tuple dataset $\{X, C, Y\}$ where $X \in \mathbb{R}^m$, $C \in \mathbb{R}^k$, $Y \in \mathbb{R}^n$ and m, k, n correspond to the dimensionalities of the image, concept and label spaces respectively. Each prediction is of the form $\hat{y} = f(g(x))$ where $g\colon X \mapsto C$ (e.g. *bird* image $\rightarrow$ $\{$*white body*, *flat yellow bill*, $\ldots$, *orange legs*$\}$) is the concept encoder, and $f\colon C \mapsto Y$ (e.g. $\{$*white body*, *flat yellow bill*, $\ldots$, *orange legs*$\}$ $\rightarrow$ *Duck*) is an interpretable predictor network.

Formal Concept Analysis [4]: Given a formal context $\langle G, M, I \rangle$, where G is a set of objects, M is a set of attributes and $I \subseteq G \times M$ is the binary relation indicating which attributes are present in which objects, a formal con-

Figure 2. Example formal concepts: intent (attributes) - extent (classes) set pairs, from the formal concept lattices built on the Imagenet100 and AWA2 datasets along with some example images from those classes. The level in the lattice that the formal concept belongs in given at the top left.

cept 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 aren't arbitrary subsets, but they are subsets of objects and attributes that have concept-forming operators defined on them ($\uparrow, \downarrow$) [4]. In simple terms, A contains objects sharing all attributes from B and B contains attributes shared by all objects from A.

$$A \subseteq G, B \subseteq M \text{ and } A^{\uparrow} = B, B^{\downarrow} = A$$
$$A^{\uparrow} = \{y \in Y \,|\, \forall x \in A : \langle x, y \rangle \in I\},$$
$$B^{\downarrow} = \{x \in X \,|\, \forall y \in B : \langle x, y \rangle \in I\}.$$

The set of all formal concepts derived from a formal context form 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, $\langle A_1, B_1 \rangle \leq \langle A_2, B_2 \rangle$ if, $A_1 \subseteq A_2$ and $B_1 \supseteq B_2$, where $\leq$ represents subconcept-superconcept ordering. This implies that abstract concepts have lesser attributes and more objects and specific concepts have more attributes and lesser objects. For example, $\langle \{dog, cat\}, \{whiskers, fur\} \rangle$ is more abstract than $\langle \{cat\}, \{whiskers, retractable\ claws, fur, wide\ eyes\} \rangle$. From this partial order, we can construct a lattice of formal concepts. This gives us a formal concept hierarchy where the concepts in the higher layers are more abstract and the ones in the lower layers are more specific.

Formal Concepts in Deep Learning Models:
Our observation was that the ideas from FCA naturally translate to a concept-based models setting, where we consider the classes as G and the class-level attribute annotations as M. Since the attribute annotations are class-level, we also have access to I. Building a formal concept lattice over such data gives us structured tuples of subgroups of classes and attributes arranged according to their level of abstraction. Some examples of the formal concepts obtained from these lattices on our datasets are provided in Fig 2. Further details on lattice construction are provided in the Appendix. Note that the lattice was constructed at a class-level (not at an image-level using instance-based attribute annotations) as we wanted to ensure that the leaf nodes in the lattice are the individual classes. This would not be the case with instance-level annotations, where formal concepts indicating specific classes (a formal concept where the extent contains a single class) may or may not exist.

Using Lattice Information for Intermediate Semantics:
Based on the formal concept lattice obtained using the formal context specified above, we define our architecture. l levels in the lattice and l positions in the backbone are chosen. The idea is to learn concepts of the degree of abstraction of level l_i at position l_i in the network. We do this by introducing l concept encoder (g_i) and predictor (f_i) layer pairs at the chosen positions. Consider level l_i in the lattice which contains a set of formal concepts c_i. In order to determine the set of attributes to learn at position l_i in the network (i.e. the set of attributes g_i learns), we union the intents of these c_i formal concepts. This gives us the set of all attributes occuring at level l_i.

$$k_i = \bigcup_{\texttt{fc} \in c_i} \texttt{fc}.intent \tag{1}$$

where, $\texttt{fc}$ is a formal concept in c_i. So, g_i then learns a mapping from $\mathbb{R}^{p_i} \rightarrow \mathbb{R}^{|k_i|}$, where p_i is the dimensionality of the feature space at position l_i in the network and $|k_i|$ is the size of the union of intents at level l_i in the lattice, i.e. the size of the attribute layer at position l_i in the network. We similarly compute the union of the intents present in the other chosen $l-1$ levels, which gives us the information of what attributes to learn at what positions in the network. This is illustrated in Fig 1. The choice of the l lattice levels and positions in the backbone are hyperparameters in our architecture. The impact of these choices is studied in the Appendix.

Leveraging Class Grouping Information for Hierarchical Learning:
The formal concept lattice provides rich class grouping information at each formal concept, which we use for hierarchical guidance. The aim is to iteratively refine the predicted classes. Our hypothesis was that the earlier layers of a network learn more abstract attributes which are perhaps not enough to distinguish between all n classes, but they are likely enough to distinguish between certain groups of classes.

Let's say that the ground truth class for the current sample is c_3 and we wish to obtain what group of classes it belongs to at level l_i in the lattice. Say level l_i has the following formal concepts:

MODEL	IMAGENET100	MODEL	AWA2
Vanilla CBM	69.39 $\pm$ 0.93	**Vanilla CBM**	80.52 $\pm$ 0.53
Posthoc CBM	67.58 $\pm$ 0.45	**Posthoc CBM**	80.87 $\pm$ 0.52
2-FCA-CBM_(5, 1)_(3, 4)	88.72 $\pm$ 0.52	**2-FCA-CBM_(3, 1)_(3, 4)**	88.44 $\pm$ 0.28
3-FCA-CBM_(7, 5, 1)_(2, 3, 4)	88.46 $\pm$ 0.03	**3-FCA-CBM_(4, 3, 1)_(2, 3, 4)**	87.61 $\pm$ 0.04

Table 1. Classification accuracy results on Imagenet100 and AWA2 datasets averaged over 3 seeds. Bottom two rows highlighted in green indicate our models. Our models have the naming scheme: `<l>-FCA-CBM_<levels>_<positions>`.

$\langle\{c_1, c_3\}, \{attr_1, attr_5\}\rangle$, $\langle\{c_2, c_4, c_5\}, \{attr_1, attr_2\}\rangle$, $\langle\{c_3, c_6\}, \{attr_4, attr_6, attr_7\}\rangle$. We consider all the classes c_3 is associated with, i.e. the other classes in a formal concept extent that c_3 is also a part of, to have an equally likely chance of getting a high activation using the current set of attributes (described in Alg A2). The union of all these classes becomes our "ground-truth group". In the provided example, these would be c_1 and c_6. As we progress through the levels, the class group sizes decrease (as concepts get more specific), the last level of which contains one class per formal concept.

In order to impose this group based iterative refinement, we multiply the post sigmoid activations of the classifier at l_i with the activations of the classifier at l_{i+1}. This is so that the models are directed to focus on the group predicted by the previous layer and to subsequently refine that group (indicated by the '$\times$' between the classifiers in Fig 1).

Overall Training Process:
Overall, our models are trained for attribute prediction and classification using iterative refinement. At each attribute layer, we use a binary cross-entropy loss ($\mathcal{L}_{BCE}$). We use a binary cross-entropy loss on all the classifiers till the penultimate classifier as well - on the ground truth groups obtained per level using the procedure outlined above. Finally for the last classifier, we use a standard cross-entropy loss ($\mathcal{L}_{CE}$).

$$\mathcal{L} = \alpha \sum_{j=1}^{l} \mathcal{L}_{BCE_j} + \beta \sum_{j=1}^{l-1} \mathcal{L}_{BCE_j} + \mathcal{L}_{CE_l} \qquad (2)$$

where, α and β are weighting hyperparameters, set to 0.01 in all our experiments.

4. Experiments

Datasets: We study our method on two benchmark datasets: Imagenet100, which is a subset of the Imagenet dataset [20], and Animals with Attributes (AWA2, [25]). AWA2 is a class-level expert-annotated dataset, while for Imagenet100, we acquire class-level attribute annotations from an LLM following the procedure outlined by [16]. Further dataset details are provided in the Appendix.

Baselines: We compare our approach with two well-known concept-based learning models: (1) Vanilla CBMs [11] and (2) Posthoc CBMs [26]. Vanilla CBMs introduced the usage of soft concepts in the learning process of a model for ante-hoc interpretability. Posthoc CBMs propose a method for converting a blackbox model into a CBM using concept activation vectors [9].

Results: Tab 1 shows the results of our study on Classification Accuracy (Acc). All models were run on a single NVIDIA GeForce RTX 3090. We use a ResNet18 network for all the AWA2 experiments and a ResNet50 network for all the Imagenet100 experiments. Our models use the following naming scheme: `<l>-FCA-CBM_<levels>_<positions>`, where l indicates the number of attribute-classifier layers placed in the network, `levels` and `positions` are both lists of l comma-separated numbers indicating the chosen level ids from the lattice and the chosen positions (here ResNet block ids) after which the attribute-classifier layer pairs are placed, respectively. Lattice level indexing is done bottom-up, i.e. level 1 indicates the leaf nodes having individual classes as formal concepts, which we always place after the final block (block 4). We experiment with 2 and 3 FCA-CBM models and observe that our models consistently outperform the baselines by significant margins. Further analysis on our models is provided in the Appendix.

5. Conclusion

In this work, we presented a method to train a neural network model to learn intermediate semantics. We present the viewpoint of considering the class and attribute information (present in the attribute-annotated datasets used for concept-based learning) as the objects and attributes of a formal context. Various aspects of the lattice constructed from this context are used for supervision. The attribute grouping information at various levels of the lattice is used to supervise attribute learning at selected positions in the network. The class grouping information is used for the iterative refinement of the classification. We observe that our models perform much better than the baselines indicating the benefit of augmenting the learning process with structured knowledge. Although there are more aspects of these models to explore, we see this as an initial effort towards the broader goal of building models more aligned with human reasoning.

References

[1] Jeffrey R Binder, Rutvik H Desai, William W Graves, and Lisa L Conant. Where is the semantic system? a critical review and meta-analysis of 120 functional neuroimaging studies. *Cerebral cortex*, 19(12):2767–2796, 2009. 1

[2] Dominik Durrschnabel, Tom Hanika, and Maximilian Stubbemann. Fca2vec: Embedding techniques for formal concept analysis. In *Complex Data Analytics with Formal Concept Analysis*, 2019. 2

[3] Octavian Ganea, Gary Becigneul, and Thomas Hofmann. Hyperbolic entailment cones for learning hierarchical embeddings. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1646–1655. PMLR, 2018. 2

[4] Bernhard Ganter and Rudolf Wille. *Formal concept analysis: mathematical foundations*. Springer Nature, 2024. 1, 2, 3

[5] Eleonora Giunchiglia and Thomas Lukasiewicz. Coherent hierarchical multi-label classification networks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2020. Curran Associates Inc. 2

[6] Eleonora Giunchiglia and Thomas Lukasiewicz. Multi-label classification neural networks with hard logical constraints. *J. Artif. Int. Res.*, 72:759–818, 2022. 2

[7] Croix Gyurek, Niloy Talukder, and Mohammad Al Hasan. Binder: Hierarchical concept representation through order embedding of binary vectors. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 980–991, 2024. 2

[8] Glyn W Humphreys and Emer ME Forde. Hierarchies, similarity, and interactivity in object recognition:"category-specific" neuropsychological deficits. *Behavioral and brain sciences*, 24(3):453–476, 2001. 1

[9] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *Proceedings of the 35th International Conference on Machine Learning*, pages 2668–2677. PMLR, 2018. 4

[10] Eunji Kim, Dahuin Jung, Sangha Park, Siwon Kim, and Sung-Hoon Yoon. Probabilistic concept bottleneck models. *ArXiv*, abs/2306.01574, 2023. 2

[11] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR, 2020. 1, 2, 4

[12] Liulei Li, Wenguan Wang, and Yi Yang. Logicseg: Parsing visual semantics with neural logic learning and reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4122–4133, 2023. 2

[13] Christian Lindig and Gartner Gbr. Fast concept analysis. *Working with Conceptual Structures - Contributions to ICCS 2000*, 2000. 1

[14] Emanuele Marconato, Andrea Passerini, and Stefano Teso. Glancenets: Interpretabile, leak-proof concept-based models. In *Neural Information Processing Systems*, 2022. 2

[15] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 2

[16] Tuomas Oikarinen, Subhro Das, Lam M Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. *arXiv preprint arXiv:2304.06129*, 2023. 2, 4, 1

[17] Konstantinos P. Panousis, Dino Ienco, and Diego Marcos. Coarse-to-fine concept bottleneck models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 2

[18] Siqi Peng, Hongyuan Yang, and Akihiro Yamamoto. Bert4fca: A method for bipartite link prediction using formal concept analysis and bert. *Plos one*, 19(6):e0304858, 2024. 2

[19] Sebastian Rudolph. Using fca for encoding closure operators into neural networks. In *International Conference on Conceptual Structures*, pages 321–332. Springer, 2007. 2

[20] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115 (3):211–252, 2015. 4, 1

[21] Yoshihide Sawada and Keigo Nakamura. Concept bottleneck model with additional unsupervised concepts. *IEEE Access*, 10:41758–41765, 2022. 2

[22] Sanchit Sinha, Mengdi Huai, Jianhui Sun, and Aidong Zhang. Understanding and enhancing robustness of concept-based models. In *AAAI Conference on Artificial Intelligence*, 2022. 2

[23] Ao Sun, Yuanyuan Yuan, Pingchuan Ma, and Shuai Wang. Eliminating information leakage in hard concept bottleneck models with supervised, hierarchical concept learning. *arXiv preprint arXiv:2402.05945*, 2024. 2

[24] Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. Order-embeddings of images and language. *arXiv preprint arXiv:1511.06361*, 2015. 2

[25] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. 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, 2019. 4, 1

[26] Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. In *The Eleventh International Conference on Learning Representations*, 2023. 2, 4

[27] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 818–833. Springer, 2014. 1

[28] Dagmar Zeithamova, Michael L Mack, Kurt Braunlich, Tyler Davis, Carol A Seger, Marlieke TR Van Kesteren, and Andreas Wutz. Brain mechanisms of concept learning. *Journal of Neuroscience*, 39(42):8259–8266, 2019. 1

Learning Hierarchically using Formal Concepts

Appendix

In this part of the paper, we provide additional details of our work, including the following information:

Table of Contents

A6. Dataset and Model Details

- **Imagenet100:** The ImageNet-100 dataset is a subset of ImageNet-1k dataset [20]. We have randomly selected 100 classes from the set 1k classes. Attributes associated to these classes are generated using LLM as described in [16]. It 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 min number of attributes active for a certain class and 25 being max.
- **AWA2:** The Animals with Attributes (AWA2) dataset [25] 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 min number of attributes active for a particular class and 39 being max.

Some example classes and their corresponding attributes are provided in Tab A4.

Model Details: We use the same backbone networks for all the baselines, i.e. ResNet50 for Imagenet100 and ResNet18 for AWA2, the same batch size of 32 and the following hyperparameters:

- *Vanilla CBMs:* All models here were trained for 150 epochs, with a learning rate of 3e-4.
- *Posthoc CBMs:* The posthoc models were trained until convergence, which was roughly 40 epochs. A learning rate of 3e-4 was used.
- *l-FCA-CBM models:* All models here were also trained for 150 epochs. A learning rate of 1e-4 was used.

A7. Lattice Details

The formal concept lattice is constructed using the `concepts` Python module. The module employs the Fast Concept Analysis algorithm [13] for generating the lattice. 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 A1).

Algorithm A1 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**
      level$[v] = \max\{\text{level}[v], \text{level}[u] + 1\}$
    end for
  end for
  return level

Algorithm A2 GET GROUND TRUTH GROUP(y, c_i):

Require: Ground truth class y, formal concept set c_i at level l_i.
  Initialize gt_group $\leftarrow \{\}$
  for fc in c_i **do**
    if y in fc.extent **then**
      for class in fc.extent **do**
        gt_group $\leftarrow$ gt_group $\cup$ {class}
      end for
    end if
  end for
  return gt_group

- The **ImageNet100** 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 are 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.
- 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.

Some examples of the formal concepts derived from the lattices of the Imagenet100 and AWA2 datasets in Fig A3.

Lattice Level Selection Heuristic: The lattice levels chosen are hyperparameters. We based our choice on the average number of class activations per level (provided for both datasets in Fig A4). In other words, this is the average ground truth group size per level. In order to calculate it, we get the ground truth group size per class and average

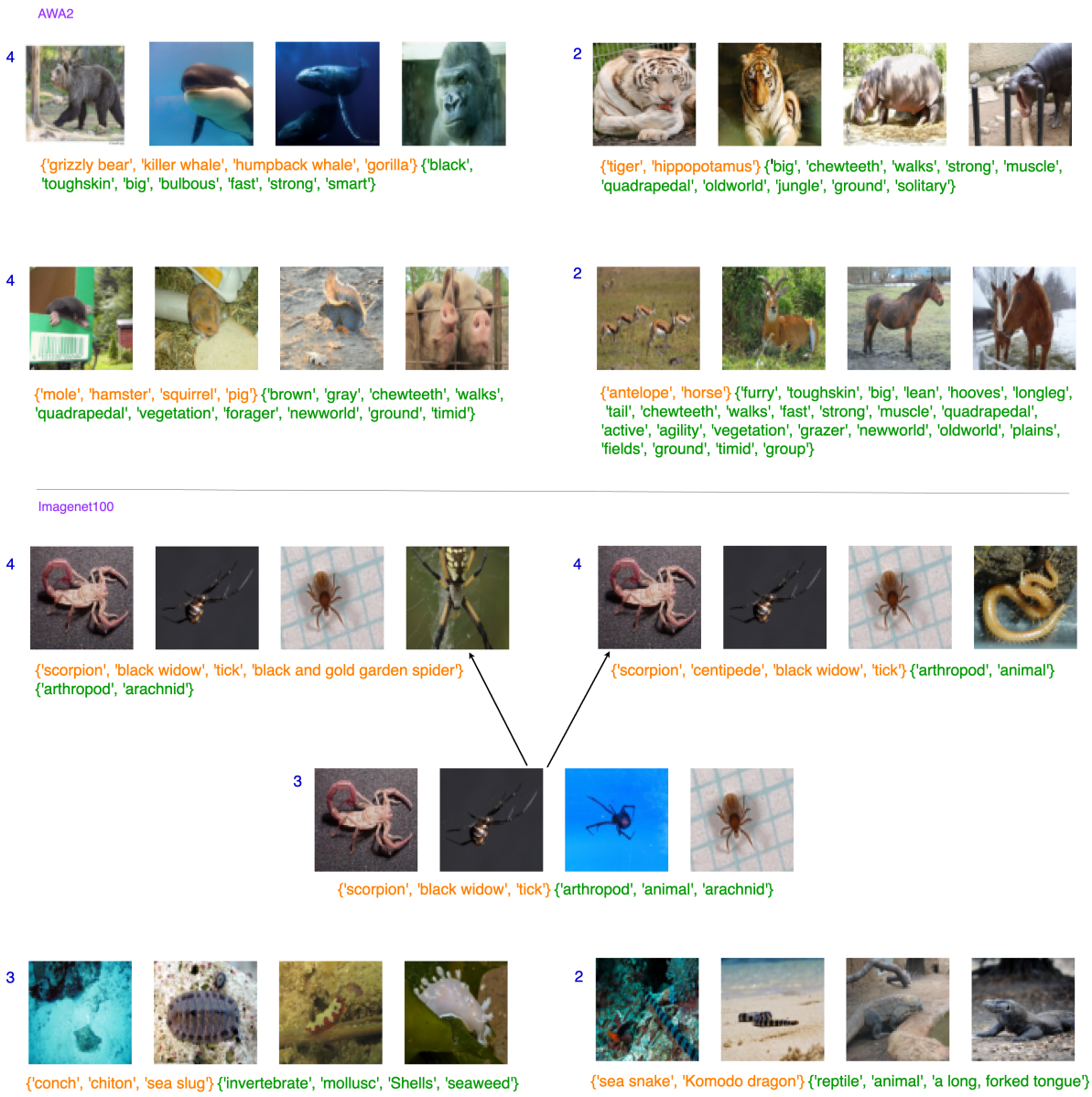

Figure A3. More examples of the formal concepts from the lattices built for the Imagenet100 and AWA2 datasets. The level that the formal concept belongs to is provided at the top left of each formal concept. An example of a parent-children relation is also provided for the Imagenet100 dataset indicated by the arrows.

it across all classes per level. Our level selection heuristic was to choose levels that have a reasonable difference in this value, while avoiding levels having too high an average class activation (*e.g.* values close to 100 for Imagenet100).

A8. Further Analysis

Position of the Intermediate Semantic Layers: We study the impact of the position of the attribute-classifier layers in our models. We vary the position of these layers in our 2-FCA-CBM models while keeping the lattice levels chosen the same and see that the position of these layers does have a marked impact on accuracy. The accuracies of both classifiers in the model are provided in Tab A2 and we see that there is a significant gain in performance by placing our first layer after block 3.

Impact of Lattice Level Choice: We also study the impact of lattice level choice on our 2-FCA-CBM model's performance. The accuracies of both classifiers on different lattice levels are provided in Tab A3, while keeping the backbone positions the same. The results indicate how different levels carry different amounts of information.

IMAGENET100			AWA2		
MODEL	CLF0	CLF1	MODEL	CLF0	CLF1
2-FCA-CBM(5,1)_(2, 4)	77.94	88.65	2-FCA-CBM_(3, 1)_(2, 4)	86.75	88.21
2-FCA-CBM(5,1)_(3, 4)	**90.29**	**89.27**	2-FCA-CBM_(3, 1)_(3, 4)	**92.95**	**88.73**

Table A2. Accuracy results on varying `intsem` layers position in the network on the Imagenet100 and AWA2 datasets.

IMAGENET100			AWA2		
MODEL	CLF0	CLF1	MODEL	CLF0	CLF1
2-FCA-CBM_(3, 1)_(3, 4)	**92.49**	88.68	2-FCA-CBM_(3, 1)_(3, 4)	92.95	88.73
2-FCA-CBM(5, 1)_(3, 4)	90.29	**89.27**	2-FCA-CBM_(6, 1)_(3, 4)	**98.02**	**88.96**

Table A3. Accuracy results with different lattice levels on the Imagenet100 and AWA2 datasets.

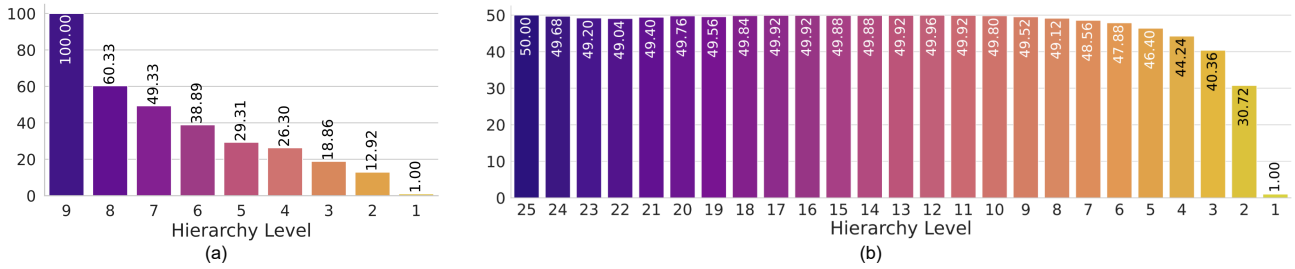

Figure A4. The average number of classes active per level for the Imagenet100 (a) and AWA2 (b) lattices.

Dataset	Class	Concepts
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
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

Table A4. Some sample classes and a subset of their corresponding attributes from the Imagenet100 and AWA2 datasets.