Abstract

Self-explaining deep models are designed to learn the latent concept-based explanations implicitly during training, which eliminates the requirement of any post-hoc explanation generation technique. In this work, we propose one such model that appends an explanation generation module on top of any basic network and jointly trains the whole module that shows high predictive performance and generates meaningful explanations in terms of concepts. Our training strategy is suitable for unsupervised concept learning with much lesser parameter space requirements compared to baseline methods. Our proposed model also has provision for leveraging self-supervision on concepts to extract better explanations. However, with full concept supervision, we achieve the best predictive performance compared to recently proposed concept-based explainable models. We report both qualitative and quantitative results with our method, which shows better performance than recently proposed concept-based explainability methods. We reported exhaustive results with two datasets without ground truth concepts, i.e., CIFAR10, ImageNet, and two datasets with ground truth concepts, i.e., AwA2, CUB-200, to show the effectiveness of our method for both cases. To the best of our knowledge, we are the first ante-hoc explanation generation method to show results with a large-scale dataset such as ImageNet.

1. Introduction

Recent years have seen an exponentially increasing interest in explainability of decisions of Deep Neural Network (DNN) models across domains including biometrics, healthcare, autonomous navigation and many more. Existing efforts in computer vision including occlusion-based, gradient-based and Shapley value-based efforts largely perform post hoc analysis [23, 27, 36], of an already trained model to identify what a DNN model looked at in an input image while making a prediction. While this is useful, the separation of explanation from prediction is not ideal. When an explanation goes wrong, it is not trivial to understand if the explanation method is incorrect, or if the model itself relied on spurious correlations to make a prediction. This has paved the need for ante hoc methods that jointly learn to explain and predict, and thus learn inherently interpretable models.

Efforts on envisioning interpretable learning models by Rudin [25] and Lipton [18] have stressed on the importance of implicitly interpretable methods over post hoc explanations in elaborate terms. In a more recent exposition, Rudin et al [26] identified ten challenges of interpretable machine
learning, which also highlighted the need for placing constraints into models to learn with better interpretability during training itself. The last few years have seen a few efforts in antehoc methods that explain through concepts, which are learned during the training of the DNN itself such as Self-Explaining Neural Networks [1], Concept Bottleneck Models [14], Concept-based Model Extraction [12], and Concept Whitening [2]. Learning concepts during training provides a natural pathway for antehoc explanations that are global (concepts that are most activated on a dataset or a class) or local (concepts that are most activated for prediction on given input image). Existing methods however either require concept-level supervision to train the model [14], or require a significant number of additional parameters in the network [1], which prohibits their use in deeper models more commonly used in practice.

In this work, we propose a new method towards learning antehoc explanations via concepts, that: (i) can be added easily to existing backbone classification architectures with minimal additional parameters; (ii) can provide explanations for model decisions in terms of concepts for an individual input image or for groups of images; and (iii) can work with different levels of supervision, including no concept-level supervision at all. This is achieved by an architectural modification added to a backbone network along with additional loss terms that allow such antehoc learning. Importantly, we show that our framework allows learning of concepts with no supervision, self-supervision as well as full supervision at a concept-level. An overview of our proposed model is shown in fig.1.

Our key contributions in this work can be summarized as follows:

• We propose a simple and effective method that jointly learns to predict and explain (through concepts) in an antehoc manner (i.e. learning to explain during training itself, as opposed to post hoc explainability methods popularly used today).

• Our method can learn to explain through concepts with different levels of supervision: (i) with no concept-level supervision; (ii) through weak supervision (self-supervised learning of concepts); as well as (iii) with concept-level supervision.

• We perform a comprehensive suite of experiments to study accuracy and explainability of our method on multiple benchmark datasets quantitatively and qualitatively, and show ablation studies on different choices made in the method. In this context, we introduce a metric based on concept intervention for antehoc explainable models such as ours.

• Our method outperforms existing methods on accuracy and explainability metrics, and achieves these results with negligible computational overhead over baseline models with no explanation component.

2. Related Work

The main objective of concept learning is to obtain a lower-dimensional representation that faithfully explains the downstream tasks, such as - object classification.

Unsupervised Concept Learning: Most of the existing methods generate meaningful explanations in an unsupervised manner, i.e., when ground truth concepts are not available for the dataset. Such methods either work as a post-hoc approach on a trained model [13] or learn an inherently interpretable model [1, 33]. TCAV [13] leverages the directional derivatives with intermediate model features to quantify the importance of a user-defined concept towards final model predictions. Though this method doesn’t require full concept annotations, the explanations are generated based on the prior knowledge of concepts over the data points. Zhou et al. [38] proposed a method to decompose the model prediction in terms of projections onto concept vectors using the model generated saliency by CAM [37]. Another recent method [33] leverages Shapley values to quantify the sufficiency of a set of concepts in explaining the model predictions through completeness measure of the concepts. Being a post-hoc explainability method, it works on trained deep networks. Unlike our method, it doesn’t allow a user to intervene on the concepts to explore the interactions between concepts and the class predictions.

The first fully-unsupervised antehoc concept learning method, SENN [1], employs a concept encoder $h(x)$ with corresponding relevances $\theta(x)$ for an image $x$ and outputs the final logit as $\theta(x)^T h(x)$. SENN is trained, following a joint training approach, with cross-entropy loss for the logits and a stability loss to enforce closeness of the similar concept relevances i.e. $\theta(x)$. Similar to SENN, our method also uses a concept encoder to extract concepts. But, we replace the heavy relevance network with a couple of simple, fully connected networks that generate explanations and perform classification.

Supervised Concept Learning: Methods such as concept bottleneck models (CBM) [14] divides the complete model into two parts. The first part is a function $g : X \rightarrow C$, that generates an intermediate concept representation $c$ from an image $x$, which is followed by the label predictor part $f : C \rightarrow Y$ to output a class label from $c$. The model predicts a class label for an image $x$ by computing $f(g(x))$. This model is trained with both concept and class label supervision, either training individual parts sequentially or both parts jointly. Kazhdan et al. proposed CME [12], a post-hoc data-efficient version of CBM, that captures intermediate representations from a pre-trained model to improve the sensitivity to the dependence between the concepts and the final prediction. Concept whitening (CW) [2] proposed a method to plug an intermediate layer in place of the batch-normalization layer of any pre-trained CNN model that helps in concept extraction by constraining the
latent layer output to represent a target concept. As opposed to CBM, we decouple the process of generating explanations and predictions. This helps us to learn concept-based explanations without losing much in predictive performance and enables the user to use the model with different levels of supervision.

Self-Supervised Concept Learning: Different self-supervised methods have been proposed to help learn better representations and boost classification accuracy. Tasks such as predicting the relative position of image patches [5], predicting rotation angle [9], recovering color channels [34], solving jigsaw puzzle games [20], and discriminating images created from distortion [6] have been extensively used in recent years. Another class of methods reconstruct images from corrupted versions or just part of it such as denoising autoencoders [28], image inpainting [21], and split-brain autoencoder [35]. Contrastive learning is another paradigm where representations are learned in such a way that similar data points are brought closer, and dissimilar data points are pushed further away [29] in the representation space. Predicting natural ordering or topology of data has also leveraged as pretext tasks in video-based [8,19,30], graph-based [11,32], and text-based [4,22] self-supervised learning. While self-supervision has been used to learn better model representations, their utility for learning concept-based explanations hasn’t been explored in the past. In our work, we explore how self-supervision can be used for learning better concept-based explanations.

3. Method

Let \( \mathcal{X} \) denote the input space, and \( \mathcal{Y} \) the output space, we assume that the training instances (or examples) \( D = \{x_i, y_i\}_{i=1}^N \) are sampled i.i.d from the source distribution \( P \) defined over \( \mathcal{X} \times \mathcal{Y} \). We also assume that \( \mathcal{X} = \mathbb{R}^d \), and \( \mathcal{Y} = \{y \in \{0, 1\}_M \mid \sum_{j=1}^M y^k = 1 \} \), where \( M \) is the number of classes, and \( y \) is an one-hot encoded vector.

We propose a generic framework to incorporate antecedent explanation (or self-explanation) modules into existing deep learning pipelines. In this paper we demonstrate it for a classification task. In practice, for classification tasks we learn a Deep Neural Network \( f_\theta = \{\eta_\theta(\cdot), g_\theta(\cdot)\} \) which consists of a base encoder (or a feature extractor) \( \eta_\theta(\cdot) \), that extracts the representation the representation vectors which are fed into a classifier function \( g_\theta(\cdot) \) (a classifier function takes the latent representation \( z = \eta_\theta(x_i) \), and then predicts the label). Typically the base encoder & the classifying function are trained together by optimizing for \( \theta = \{\theta_e, \theta_c\} \) such that the output of the network \( \hat{y}_i = f_\theta(x_i) \) minimizes a loss \( L_C(\hat{y}_i, y_i) \) over the set of training instances \( D \).

To incorporate implicit learning of interpretable concepts, in addition to the existing components of classical classification pipelines described previously, we introduce a concept encoder \( \Psi_{\theta_c}(\cdot) \) which takes the representation \( \eta_{\theta_e}(\cdot) \) and learns a set of interpretable concepts \( \{\psi^1, \ldots, \psi^C\} \) (where \( C \) is the number of concepts), to explain the predictions provided by \( f_\theta \). In general, concepts are low dimensional representation that can be characterized as \( C \in \mathbb{R}^{K \times d} \), i.e., every concept \( c \in \mathbb{R}^d \) belongs to one of the total \( k \leq K \) concepts. In our work, we learn one-dimensional concepts, i.e., our setup uses \( k \) concepts, with every concept is represented by a scalar value.

To encourage the model to learn concepts \( \{\psi^1, \ldots, \psi^C\} \) that capture the semantics of the input image \( x_i \), we pass the concepts to a decoder \( h_{\theta_c}(\cdot) \) which reconstructs the image \( \hat{x}_i \). We then add a loss \( \mathcal{L}_R(x_i, \hat{x}_i) \) which measures the reconstruction error to the overall loss \( \mathcal{L} \). \( \mathcal{L}_R \) penalises the model \( f_\theta \), if the concepts aren’t suffice to generate an accurate reconstruction \( \hat{x}_i \) of the input image \( x_i \). In our paper, we use an \( L_2 \) loss.

Since the concepts \( \{\psi^1, \ldots, \psi^C\} \) explain the prediction of a DNN \( f_\theta \). Ideally, they should be informative enough by themself to predict the input instance \( x_i \) correctly. To enforce that the learnt concepts not only explain the prediction but are also informative, we penalize the model \( f_\theta \), if the predictions \( s_{\theta_{ce}}(\Psi_{\theta_c}(x_i)) \) (where \( s_{\theta_{ce}} \) is a classification function which predicts the class labels taking the concepts as input) based on the concepts \( \{\psi^1, \ldots, \psi^C\} \) and prediction by the DNN \( f_\theta \) differs. We enforce that the concepts learned should be individually informative by adding a fidelity loss \( \mathcal{L}_F \) to the overall loss \( \mathcal{L} \).

Taking the proposed modifications into consideration, the overall loss \( \mathcal{L}_O \) of the model can be written as follows:

\[
\mathcal{L}_O = \mathcal{L}_C(y_i, \hat{y}_i) + \alpha \mathcal{L}_R(x_i, \hat{x}_i) + \beta \mathcal{L}_F(f_\theta(x_i), s_{\theta_{ce}}(\Psi_{\theta_c}(x_i)))
\]

In practice, most data sets seldom include annotations of concepts (or attributes) that could be used to learn a self-explaining model. However, few exceptions contain concepts (or attributes) that the models can leverage while learning to explain their predictions. The majority of existing frameworks either work when only annotation of concepts is available, or data sets don’t contain any additional annotations, but not both. Often it is neither trivial nor efficient to incorporate alternate forms for supervision in these existing frameworks. In comparison, our framework offers the flexibility to incorporate different forms of supervision whenever available easily. To illustrate this, we demonstrate how to incorporate i) complete supervision (supervised learning of interpretable concepts), ii) zero-supervision (unsupervised learning of interpretable concepts), and iii) a weaker form of supervision that is cheaply available like self-supervision.

By default, our framework works with data sets where the annotation of concepts isn’t available. In cases when they are available, we can easily incorporate them into the learning process by adding a loss \( \mathcal{L}_E(\Psi_{\theta_c}(x_i), a_{x_i}) \).
(where $a_{x_i}$ is the concept (or attribute) annotation of $x_i$). $L_E(\psi_{\theta_{ss}}(x_i), a_{x_i})$ would penalize the model if the concepts learned aren’t similar to the annotation in the data set for the corresponding instance. We then train the model by optimising $\theta$ such that the output of the network $\tilde{y}_i = f_\theta(x_i)$ minimizes a loss $L_O + \mu L_E$ over the training set.

Even when direct supervision isn’t available for concepts, it is possible to learn a robust set of high fidelity interpretable concepts $\{\psi^1, \ldots, \psi^C\}$ by leveraging the underlying structure of the data by incorporating supervisory signals obtained directly from the data itself. This technique is popularly known as self-supervision. In our framework, we incorporate self-supervision as an auxiliary task with a loss $L_{SS}$, and the auxiliary task shares the parameters with our model until the concept encoder $\psi_{\theta_{ss}}(.)$. In this paper, we choose rotation prediction as an auxiliary task. The task involves rotating the image by one of 0, 90, 180, or 270 degrees and predicting the rotation angle $r_i$ as a four-way classification problem through an auxiliary head. We can also easily incorporate other self-supervision tasks into our framework.

As opposed to existing techniques where the branch for the auxiliary task uses the output of feature extractor (or base encoder) for the self-supervision tasks, in our case, we use the output of the concept encoder $\psi_{\theta_{ss}}(.)$. In turn, this helps us to ensure that the set of interpretable concepts $\{\psi^1, \ldots, \psi^C\}$ always respects the underlying structure of the data and has high fidelity. To estimate $L_{SS}$ we pass the output of the concept encoder $\psi_{\theta_{ss}}(.)$ through a classifier function $\zeta_{r_{\theta_{ss}}}(.)$ that predicts the angle of rotation, we then compute cross-entropy between $\zeta_{r_{\theta_{ss}}}(.)$ and $r_i$. Like in other cases, we jointly train the model and the auxiliary head by optimising $\theta$ such that the output of the network $\tilde{y}_i = f_\theta(x_i)$ minimizes a loss $L_O + \gamma L_{SS}$ over the training set. In cases where ground truth annotations of concepts aren’t available, and an auxiliary self-supervision task isn’t used $\mu$, and $\gamma$ are respectively set to 0.

$$L_O' = \mu L_E(\psi_{\theta_{ss}}(x_i), a_{x_i}) + \gamma L_{SS}(r_i, \zeta_{r_{\theta_{ss}}}(r_i))$$

Even though our framework incorporates additional components to existing deep learning backbones (or pipelines), we can discard most of them after the training. We only retain the sub-network (or module) to generate explanations in addition to the ones on standard deep learning pipelines (i.e., feature extractor and the classifier function) during the prediction time. Hence, compared to existing self-explaining models, the additional cost incurred by our framework is relatively insignificant.

4. Experiments

We show that our framework achieves competitive predictive accuracy compared to standard classification pipelines, as well as meaningful explanations. We report results with our method on CIFAR10, ImageNet, AwA2 and CUB-200 with different levels of concept supervision according to availability of ground truth concepts i.e. unsupervised manner on CIFAR10 [15], ImageNet [3] and with concept supervision on AwA2 [17], CUB-200 [31]. We also report results with AwA2, CUB when our model is trained without concept supervision to show the effectiveness of our method in both cases, i.e., with and without concept supervision. We consider SENN [1] and CBM [14] as our baselines considering the basic methods for unsupervised and supervised concept learning. The implementation of our method is publicly available at this link.

Dataset Details: The CIFAR-10 dataset [15] consists of 32x32 colour images in 10 classes, each with 5000 train images and 1000 test images per class. The ImageNet dataset [3] is comprised of more than 1 million images and 1000 object classes of natural images. Aw-A2 dataset [17] consists of 37322 images of total 50 animals classes with 85
numeric attribute. The other attribute dataset we considered is CUB-200 [31], an image dataset with photos of 200 bird categories with a total of 6033 images and 312 attribute annotations for each image.

**Architecture Details:** We use ResNet18 [10] as our backbone network for all datasets, as there is no standard architecture followed in the literature related to concept learning. The backbone network resembles to \( f_\theta = \{ \eta_e(\cdot), g_\theta(\cdot) \} \) as given in Sec.3. The output of the feature encoder \( \eta_e(\cdot) \) is also passed to the concept encoder \( \Psi_{\theta_c}(\cdot) \) which is a single fully connected layer, that outputs a set of interpretable concepts \( \{ \psi^1, \ldots, \psi^C \} \) where \( C \) is the number of concepts. We considered 10 and 100 concepts for CIFAR10 and ImageNet respectively. The number of concepts (or attributes) for AwA2 and CUB-200 is 85 and 312, respectively. We kept the number of concepts the same for fair comparison while training our model with these datasets for both unsupervised concept learning and learning with concept supervision. The classification function \( s_{\theta_{ce}}(\cdot) \) that predicts the class labels, taking the concepts as input, is also a single fully connected layer. The number of parameters of the concept encoder and the classification network, taking the concepts as inputs, vary for different datasets based on the number of concepts and classes. We implement the decoder \( h_{\theta_d}(\cdot) \) as a set of deconvolution layers.

**Storage and Time Complexity:** The architecture proposed by SENN requires a vast number of parameters with both the concept and relevance encoder contributing to it. Our model alleviates this issue by removing the relevance network altogether and adding the concept classification network that serves a similar purpose. However, the decoder network is required to make the concepts capture sufficient information to reconstruct the image. Hence, our overall network requires \( \sim 60\% \) of the space and training time compared to SENN. Compared to CBM, our method requires \( \sim 1.5 \) times the space and training time. For example, the training times required for one epoch on CIFAR10 are 4.2s, 6.9s & 11.3s for CBM, Ours & SENN methods with batch size 128 in one Tesla V100 GPU. This is due to a decoder that enables our framework to support cases when concept supervision isn’t available, which CBM doesn’t. Our approach takes almost similar inference time as CBM as we don’t use the decoder network during inference and concept extraction. Please note that these storage and time measurements during training are with ResNet18 backbone architectures, and the gap with CBM will further reduce with more complex backbone networks.

**Predictive Performance:** Table 1 reports the predictive performance of our method as well as the baseline methods with CIFAR10, ImageNet, AwA2, and CUB datasets. As CBM requires concept supervision, we can’t use this method for CIFAR10 and ImageNet. An unsupervised version of our method outperforms SENN significantly for all the datasets. CBM, being a method with concept supervision, performs slightly better than our unsupervised version. Our approach, with concept supervision, beats CBM by a large margin. Please note that the predictive performance by our method, reported in table 1, is solely based on the backbone network \( f_\theta(\cdot) \). We decoupled the main prediction task and concept extraction so that our model doesn’t sacrifice much of the predictive performance and still can produce meaningful explanations.

### 4.1. Quantitative Evaluation

We evaluate and compare the concept-based explanations generated by our method with other state-of-the-art frameworks like SENN and CBM. We consider metrics of interpretability that assess the effectiveness of additional losses we use in our framework. Apart from the existing metrics such as faithfulness, fidelity, and explanation error, we also perform interventions on the generated concepts to illustrate their meaningfulness. Fig. 3 shows examples of interventions that lead to the model changing its prediction when we intervene on the top concept. Besides the predictive performance, our method consistently outperforms the baseline methods in all the other explainability metrics as explained below.

**Faithfulness Metric:** In practice, we want the concepts learned to be meaningful and faithfully explain the model’s predictions. To evaluate how faithful the explanations generated by different frameworks are, we measure the predictive capacity of the generated concepts, i.e., from the output of \( s_{\theta_{ce}}(\cdot) \), in our case. This metric represents the capability of the overall concept vector to predict the ground truth task label. It is similar to other measures such as explicitness [24] and informativeness [7] used to measure feature disentanglement.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SENN [1]</th>
<th>CBM [14]</th>
<th>Ours w/o sup</th>
<th>Ours w sup</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR10</td>
<td>84.50</td>
<td>NA</td>
<td>90.86</td>
<td>NA</td>
</tr>
<tr>
<td>ImageNet</td>
<td>58.55</td>
<td>NA</td>
<td>59.73</td>
<td>NA</td>
</tr>
<tr>
<td>AwA2</td>
<td>76.41</td>
<td>81.61</td>
<td>79.29</td>
<td>83.30</td>
</tr>
<tr>
<td>CUB-200</td>
<td>58.81</td>
<td>64.17</td>
<td>61.49</td>
<td>62.59</td>
</tr>
</tbody>
</table>

Table 2. Comparison of faithfulness (in %, predictive performance solely based on concepts) of concepts generated by different methods on CIFAR10, ImageNet, AwA2 and CUB-200 data sets. (w=with, w/o=without)
**Fidelity Metric:** Fidelity measures the fraction of the data points where the model prediction matches the prediction from the interpretation. It is widely used to measure how well the generated explanations approximate the model predictions. This metric does not apply to methods where the interpreter is directly used to provide model prediction, such as SENN and CBM. Table 3 reports all the comparative results for all the datasets. We use fidelity loss $L_F$ during training which justifies the high fidelity score for our models.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>OURS w/o sup</th>
<th>OURS w sup</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR10</td>
<td>99.11</td>
<td>NA</td>
</tr>
<tr>
<td>ImageNet</td>
<td>90.22</td>
<td>NA</td>
</tr>
<tr>
<td>AwA2</td>
<td>97.84</td>
<td>97.19</td>
</tr>
<tr>
<td>CUB-200</td>
<td>97.52</td>
<td>95.87</td>
</tr>
</tbody>
</table>

Table 3. Comparison of fidelity (the % of match between model prediction and the prediction through the interpretation.) of concepts generated by different methods on CIFAR10, ImageNet, AwA2 and CUB-200 data sets. (w=with, w/o=without)

**Explanation Error:** In data sets like CUB and AwA2, where ground truth concepts are available, we also measure how close are the concepts learned to the ground truth. We compute the $L_2$ distance between the concepts learned and the ground truth concepts to measure the alignment. From table 4, we can observe that the concepts generated by our method is most aligned to the ground truth concepts. While this should be the case for methods with concept supervision, our method without concept supervision also performs better than SENN, which illustrates our method’s effectiveness in learning concept-based explanation even when annotations for concepts (or attributes) aren’t available.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SENN</th>
<th>CBM</th>
<th>OURS (w/o sup)</th>
<th>OURS (w sup)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AwA2</td>
<td>0.99</td>
<td>1.34</td>
<td>0.97</td>
<td>1.29</td>
</tr>
<tr>
<td>CUB</td>
<td>0.91</td>
<td>1.17</td>
<td>0.89</td>
<td>1.14</td>
</tr>
</tbody>
</table>

Table 4. Comparison of explanation error (we measure the mismatch using $L_2$ distance, hence lower the better) between concepts generated by different methods and the ground truth concepts (or attributes) on AwA2 and CUB-200 data sets. (w=with, w/o=without)

**Intervention on Concepts:** To study the concepts’ usefulness, we scale their values in the $[0, 1]$ range, select those above the threshold value $\omega$, set the concepts to 0, and then predict the label solely based on the intervened concept vector. A change in the prediction means that the concepts zeroed are essential for explaining the model’s decision. We repeat this procedure for all the instances in the test set and measure the predictive performance solely based on the generated concepts. A lower value indicates that the concepts generated are faithfully explaining the models’ decisions. Ideally, the predictive ability of the concepts generated by methods like SENN and CBM should be higher. Since, in their cases, the interpreter (or the explainer) is directly used to generate the model prediction, the predictive performance based on concepts generated should be lower. But, you can observe from table 5 the predictive performance after the intervention is the lowest for the proposed framework.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Baselines</th>
<th>OURS w/o sup</th>
<th>OURS w sup</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR10</td>
<td>66.57</td>
<td>43.19</td>
<td>43.19</td>
</tr>
<tr>
<td>ImageNet</td>
<td>43.91</td>
<td>34.52</td>
<td>34.52</td>
</tr>
<tr>
<td>AwA2</td>
<td>61.39</td>
<td>37.61</td>
<td>37.61</td>
</tr>
<tr>
<td>CUB-200</td>
<td>47.22</td>
<td>34.38</td>
<td>34.38</td>
</tr>
</tbody>
</table>

Table 5. The effect of interventions (accuracy in % after intervention, lower the better) on concepts generated by different methods for CIFAR10, ImageNet, AwA2 and CUB-200 data sets. (w=with, w/o=without)

**4.2. Qualitative Results**

Qualitative results are significant for methods that explain models through concept-based representations. We generate explanations corresponding to every concept as the most representative images from the dataset. We present results from CIFAR10 and ImageNet datasets in the main paper and move the rest to the Appendix due to space constraints. The top concept activations generated for ImageNet are presented in Fig.4. We can observe that every concept captures homogeneous characteristics from the dataset that mostly corresponds to a class or similar class type. For example, $\psi^7$ represents concepts of faces for cheetah and some other similar types of cat species.
Figure 4. A subset of 10 concept activations learnt by our framework on ImageNet. All these examples were correctly predicted by the model, and it can be seen that the each concept captures a certain set of homogeneous properties corresponding to a class. For ImageNet, we observe that the learned concepts are shared across the classes. For instance, $\psi^7$ is shared between tiger, cheetah, and different types of cat classes, and $\psi^6$ is shared among different forms of wolf and dog classes.

Figure 5. Effect of decoder for CIFAR10 dataset. We can see the without the decoder & the corresponding reconstruction loss the concept-based explanations (on the right) doesn’t capture any homogeneous property and are hard to understand, unlike the case where decoder is present (on the left).

For data sets like CIFAR10, where there isn’t much intersection of higher-level attributes across classes, we observe that each learned concept only corresponds to characteristics (or features) from a single class. In comparison, for data sets like ImageNet, where there is a lot of intersection between the higher-level attributes of different classes, we observe that the learned concepts are shared across the classes. For instance, $\psi^7$ is shared between tiger, cheetah, and different types of cats classes, and $\psi^6$ is shared among different types of wolf and cat classes (refer Figs 4 & 5).

4.3. Global Explanations

An advantage of concept-based explanation methods compared to others is that they provide local as well global explanations. We identify class-concept (or attribute) pairs with a high proportion of co-occurrence to generate global explanations. We consider CIFAR10 and AwA2 for our experiments to explain the effectiveness of our method in generating such global explanations on datasets without and with ground truth concepts. Simply analyzing these can reveal helpful information about the generated concepts. For instance, based on samples, we can see that (from Fig.6) concept 'ocean' is a distinguishing attribute for class killer+whale of AwA2. Similarly, $\psi^1$ represents a distinctive concept for cat class of CIFAR10 (from $\psi^1$ to $\psi^5$ of Fig.5 on the left).

5. Ablation Studies

Importance of Self-supervision: As discussed in Sec.3, our framework enables us to incorporate self-supervision on
Figure 6. Example class-attribute pair analysis on AwA2 and CIFAR10 datasets with high global relevance (proportion of co-occurrence)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Baseline</th>
<th>SENN</th>
<th>w/o sup</th>
<th>OURS w self-sup</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR10</td>
<td>84.50</td>
<td>91.68</td>
<td>90.86</td>
<td>90.93</td>
</tr>
<tr>
<td>ImageNet</td>
<td>58.55</td>
<td>58.73</td>
<td>58.70</td>
<td>60.28</td>
</tr>
<tr>
<td>AwA2</td>
<td>76.41</td>
<td>79.29</td>
<td>79.17</td>
<td></td>
</tr>
<tr>
<td>CUB-200</td>
<td>58.81</td>
<td>63.04</td>
<td>61.49</td>
<td>61.81</td>
</tr>
</tbody>
</table>

Table 6. Comparison of predictive performance (accuracy in %) of models on different methods on CIFAR10, ImageNet, AwA2 and CUB-200 data sets with and without self-supervision. (w=with, w/o=without)

coder, keeping all the other model parts unchanged. We generate the explanations of the trained model and present them in fig. 5. For comparing with the complete model (i.e., our model with decoder), we add explanations generated by our complete model in the same figure. The first and the last five columns are explanations generated by our complete model and the model without a decoder, respectively. These examples support our claim about the importance of decoder for learning better concepts. Please note that the model without a decoder performs slightly better than our complete model, but sacrificing a little bit of predictive performance can be justified to gain trust in the model.

6. Conclusion

In this work, we propose a new framework towards learning ante-hoc concept-based explanations that: (i) can be added easily to existing backbone classification architectures with minimal additional parameters; (ii) can provide explanations for model decisions in terms of concepts for an individual input image or groups of images; & (iii) can work with different levels of supervision, including no concept-level supervision at all. Even though our framework incorporates additional components to existing deep learning backbones (or pipelines), we can discard most of them after the training. We only retain the sub-network (or module) to generate explanations in addition to the ones on standard deep learning pipelines (i.e., feature extractor and the classifier function) during the prediction time. Hence, compared to existing self-explaining models, the additional cost incurred by our framework is relatively insignificant. We performed a comprehensive suite of experiments to study the accuracy and explainability of our method on multiple benchmark datasets both quantitatively and qualitatively. Our approach consistently outperforms the baseline methods in all the datasets. In addition to this, we also performed ablation studies to illustrate the importance of additional components added by our method.

Acknowledgements. This work has been partly supported by the funding received from MoE and DST, Govt of India, through the UAY and ICPS programs. We thank the anonymous reviewers for their valuable feedback that improved the presentation of this paper.
References


