A APPENDIX

A.1 BACKGROUND ON NEURO-SYMBOLIC AI

Neuro-symbolic systems are hybrid models that leverage the robustness of connectionist methods and the soundness of symbolic reasoning to effectively integrate learning and reasoning Garcez et al. (2015); Besold et al. (2017). Research to combine logic and the neural network has received renewed attention over the last few years Lamb et al. (2020); De Raedt et al. (2020); Badreddine et al. (2022). According to the taxonomy provided in Lamb et al. (2020), there are six variants of neuro-symbolic systems depending on how the neural and symbolic components of the model interact with each other. The Type 1 models are standard deep learning models where the input and output are made of symbols. For example, a machine translation model is a type 1 system that deals with words in the input and output. In type 2 systems, the neural network is loosely coupled with a symbolic component. An example is DeepMind's AlphaGo, where the symbolic component is a Monte Carlo tree search. In a Type 3 system, the neural component responsible for a specific task interacts via its input and output with a symbolic component responsible for a complimentary task (e.g Mao et al. (2019); Lu et al. (2019)). In Type 4 systems, the Neural and symbolic components of the model are more integrated, i.e. the symbolic knowledge is compiled into the training set of the neural network (*e.g.,* Arabshahi et al. (2018); Lample & Charton (2019)). In Type 5 systems, symbolic logic rules are mapped to embeddings that act as a soft constraint on the network's loss function Huang et al. (2018); Arabshahi et al. (2021b). Finally, the neural and symbolic components of the model in a Type 6 system (arguably the most capable) are fully integrated such that the model is capable of actual symbolic reasoning inside the neural component Arabshahi et al. (2021a).

A.2 DATASETS

CUB-200 The Caltech-UCSD Birds-200-2011 (Wah et al. (2011)) is a fine-grained classification dataset comprising 11788 images and 312 noisy visual concepts. The aim is to classify the correct bird species from 200 possible classes. We adopted the strategy discussed in Koh et al. (2020) to extract 108 denoised visual concepts. Also, we utilize training/validation splits shared in Barbiero et al. (2022). Finally, we use the state-of-the-art classification models Resnet-101 (He et al. (2016)) and Vision-Transformer (VIT) (Wang et al. (2021)) as the blackboxes f.

Animals with attributes2 (Awa2) AwA2 dataset Xian et al. (2018) consists of 37322 images of total 50 animals classes with 85 numeric attribute. We aim to classify the correct animal species from 200 possible classes. We use the state-of-the-art classification models Resnet-101 (He et al. (2016)) and Vision-Transformer (VIT) (Wang et al. (2021)) as the blackboxes f.

HAM10000 HAM10000 (Tschandl et al. (2018)) is a classification dataset aiming to classify a skin lesion benign or malignant. Following Daneshjou et al. (2021), we use Inception Szegedy et al. (2015) model, trained on this dataset as the blackbox f. We follow the strategy in Lucieri et al. (2020) to extract the 9 concepts from the Derm7pt (Kawahara et al. (2018)) dataset.

MIMIC-CXR We use 220,763 frontal images from the MIMIC-CXR dataset Johnson et al. aiming to classify cardiomegaly and effusion. We obtain the anatomical and observation concepts from the RadGraph annotations in RadGraph's inference dataset (Jain et al. (2021)), automatically y generated by DYGIE++ (Wadden et al. (2019)). We use the test-train-validation splits from Yu et al. (2022) and Densenet121 Huang et al. (2017) as the blackbox f.

A.3 LOSS FUNCTION

In this section, we will discuss the loss function used in distilling the knowledge from the blackbox to the symbolic model. We remove the superscript k for brevity. We adopted the optimization proposed in Geifman & El-Yaniv (2019).Specifically, we convert the constrained optimization problem in equation 2 as

$$\mathcal{L}_s = \mathcal{R}(\pi, g) + \lambda_s \Psi(\tau - \zeta(\pi)) \tag{5}$$
$$\Psi(a) = \max(0, a)^2,$$

where τ is the target coverage and λ_s is a hyperparameter (Lagrange multiplier). We define $\mathcal{R}(.)$ and $\mathcal{L}_{g,\pi}(.)$ in equations 1 and 3 respectively. ℓ in equation 3 is defined as follows:

$$\ell\big(f, g\big) = \ell_{distill}(f, g) + \lambda_{lens} \sum_{i=1}^{r} \mathcal{H}(\beta^i), \tag{6}$$

where λ_{lens} and $\mathcal{H}(\beta^i)$ are the hyperparameters and entropy regularize, introduced in Barbiero et al. (2022) with r being the total number of class labels. Specifically, β^i is the categorical distribution of the weights corresponding to each concept. To select only a few relevant concepts for each target class, higher values of λ_{lens} will lead to a sparser configuration of β. ℓ is the knowledge distillation loss Hinton et al. (2015), defined as

$$\begin{aligned}\ell(f, g) =& (\alpha_{KD} * T_{KD} * T_{KD}) KL\big(\text{LogSoftmax}(g(.)/T_{KD}), \text{Softmax}(f(.)/T_{KD})\big) + \\ & (1 - \alpha_{KD}) CE\big(g(.), y\big),\end{aligned} \tag{7}$$

where T_{KD} is the temperature, CE is the Cross-Entropy loss, and α_{KD} is relative weighting controlling the supervision from the blackbox f and the class label y.

As discussed in Geifman & El-Yaniv (2019), we also define an auxiliary interpretable model using the same prediction task assigned to g using the following loss function

$$\mathcal{L}_{aux} = \frac{1}{m} \sum_{j=1}^{m} \ell_{distill}(f(\boldsymbol{x_j}), g(\boldsymbol{c_j})) + \lambda_{lens} \sum_{i=1}^{r} \mathcal{H}(\beta^i), \tag{8}$$

which is agnostic of any coverage. $\mathcal{L}_{aux}$ is necessary for optimization as the symbolic model will focus on the target coverage τ before learning any relevant features, overfitting to the wrong subset of the training set. The final loss function to optimize by g in each iteration is as follows:

$$\mathcal{L} = \alpha \mathcal{L}_f + (1 - \alpha) \mathcal{L}_{aux}, \tag{9}$$

where α is the can be tuned as a hyperparameter. Following Geifman & El-Yaniv (2019), we also use $\alpha = 0.5$ in all of our experiments.

A.4 Algorithm

Algorithm 1 explains the overall training procedure of our method. Figure 8 displays the architecture of our model in iteration k.

Selecting the number of experts We follow two principles to stop the recursive process.
1) Each expert should have enough data to be trained reliably (coverage ζ^k). If insufficient samples fallinto the expert, we stop the process.
2) If the latest residual (r^k) is under-performing, it is not a reliable black box to distill. We stop the procedure to avoid degrading the overall accuracy.

A.5 Code availability

We will upload the code upon the decision from the reviewers.

A.6 Flow diagram to eliminate shotcut

Figure 9 shows the flow digram to eliminate shortcut.

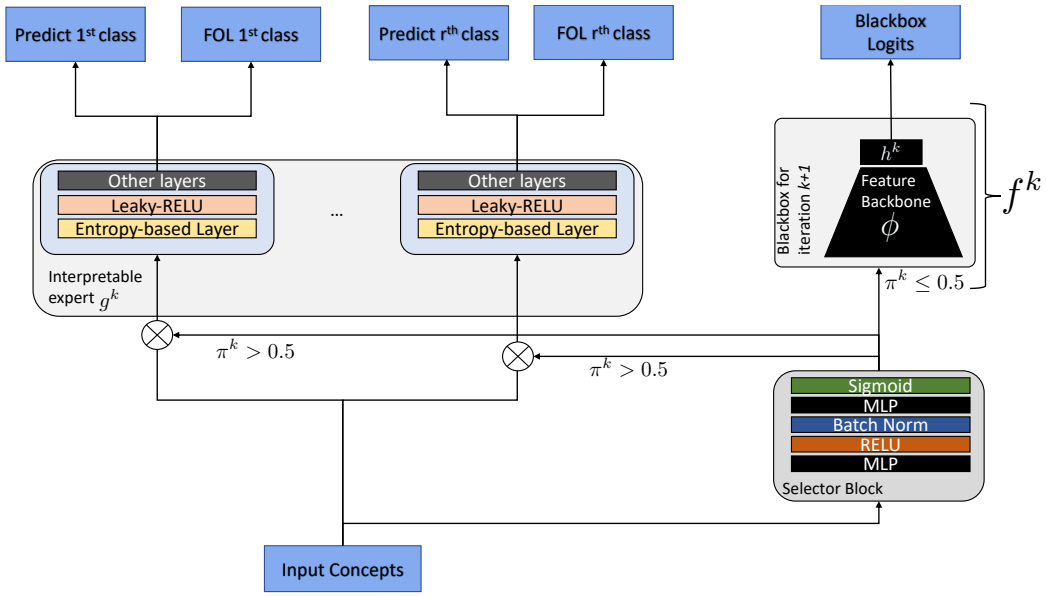

Figure 8: Architectural details of our model in an iteration k during inference. At inference, selector routes the samples to go through the interpretable expert g^k if the probability $\pi^k \geq 0.5$. If $\pi^k < 0.5$, the selector routes the samples, through f^k, the Blackbox for iteration $k+1$. Note $f^k = h^k(\Phi(.)$ is an approximation of the residual $r^k = f^{k-1} - g^k$.

Algorithm 1 Training the sparse mixture of experts to generate FOL explanations locally

Input: Training set: $\{\mathcal{X}, \mathcal{Y}, \mathcal{S}\}$; trained blackbox $f^0 = h^0(\Phi(.))$ using supervision of $\mathcal{Y}$; K as the # iterations; Coverages $\tau_1, \ldots, \tau_k$

Output: Sparse mixture of experts and their selectors $\{g^k, \pi^k\}_{k=1}^K$,

1: Fix Φ.
2: Train t by minimizing BinaryCrossEnt($t(\Phi(\boldsymbol{x}), \mathcal{S})$
3: Form a concept bank $\mathcal{C}$ with p concepts after discarding the concepts whose validation auroc (accuracy) ≤ 0.7 (70%)
4: **for** `iteration` $k = 1 \ldots K$ **do**
5: Fix $\pi^1 \ldots \pi^{k-1}$.
6: Minimize $\mathcal{L}^k$ using equation 9 to learn π^k and g^k.
7: Calculate $r^k = f^{k-1}(.) - g^k(.)$
8: Minimize equation 4 to learn $f^k(.)$, the new blackbox for the next iteration $k+1$
9: **end for**
10: **for** `experts` $k = 1 \ldots K$ **do**
11: **for** `each sample in the test-set` **do**
12: Sort the concepts according to their attention scores from different experts in descending order.
13: Initialise FOL_bucket as empty list.
14: Select one concept $\{c^i\}_{i=1}^p$ at a time from the sorted concept bank in step 12 until $g(c^i)$ = g($\boldsymbol{c}$) and add those concepts in the FOL_bucket.
15: Construct the FOL expression from FOL_bucket using Barbiero et al. (2022).
16: **end for**
17: **end for**

A.7 Architectural details of symbolic experts and hyperparameters

Table 1 demonstrates different settings to train the Blackbox of CUB-200, Awa2 and MIMIC-CXR respectively. For the VIT-based backbone, we used the same hyperparameter setting used in the state-of-the-art Vit-B_16 variant in Wang et al. (2021). To train t, we flatten the feature maps from the last convolutional block of ϕ using "Adaptive average pooling" for CUB-200 and Awa2

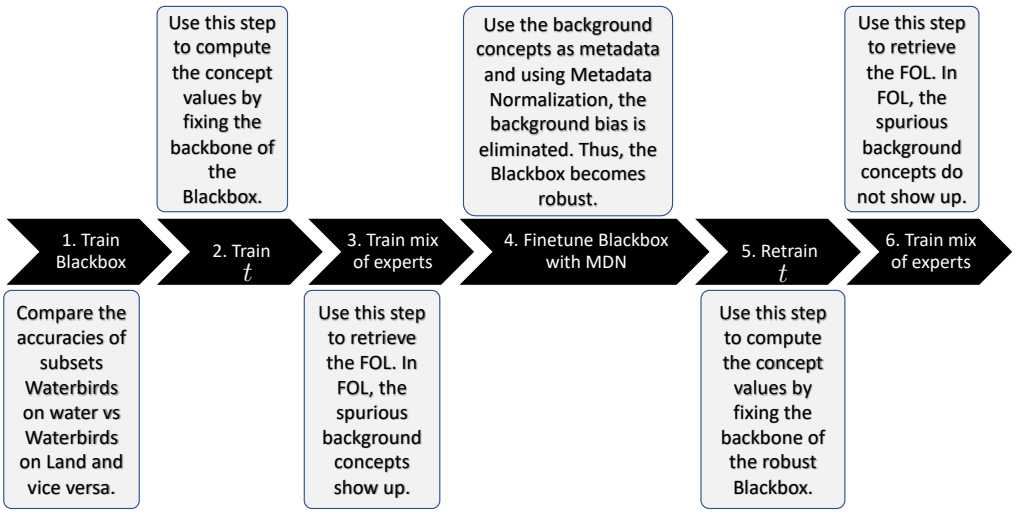

Figure 9: The flow diagram to eliminate the shortcut from vision datasets using FOL by mixture of interpretable experts.

Table 1: Hyperparameter setting of different convolution-based Blackboxes used by CUB-200, Awa2 and MIMIC-CXR

Setting	**CUB-200**	**Awa2**	**MIMIC-CXR**
Backbone	ResNet-101	ResNet-101	DenseNet-121
Pretrained on ImageNet	True	True	True
Image size	448	224	448
Learning rate	0.001	0.001	0.01
Optimization	SGD	Adam	SGD
Weight-decay	0.00001	0	0.0001
Epcohs	95	90	50
Layers used as ϕ	till 4^{th} ResNet Block	till 4^{th} ResNet Block	till 4^{th} DenseNet Block
Flattening type for the input to t	Adaptive average pooling	Adaptive average pooling	Flatten

datasets.For MIMIC-CXR and HAM10000, we flatten out the feature maps from the last convolutional block. For VIT-based backbones, we take the first block of representation from the encoder of VIT. For HAM10000, we use the same Blackbox in Yuksekgonul et al. (2022). Tables 2, 3, 4, 5 enumerate all the different settings to train the interpretable experts for CUB-200, Awa2, HAM, and MIMIC-CXR respectively. All the residuals in different iterations follow the same settings as their blackbox counterparts.

A.8 More Results

A.8.1 Summary statistics of no. of concepts used for various architectures

Figure 10 shows the summary statistics for multiclass classification vision datasets. For both datasets, we observe that the VIT-based MoIE uses fewer concepts for explanation than their ResNet-based counterparts. For example, for the CUB-200 dataset, expert6 of VIT-backbone requires 25 concepts compared to 105 by expert6 of ResNet-101-backbone (Figure 10a). The 105 concepts by

Table 2: Hyperparameter setting of interpretable experts (g) trained on ResNet-101 (top) and VIT (bottom) blackboxes for the CUB-200 dataset

Settings based on dataset	**Iteration 1**	**Iteration 2**	**Iteration 3**	**Iteration 4**	**Iteration 5**	**Iteration 6**
CUB-200 (ResNet-101)						
+ Batch size	16	16	16	16	16	16
+ Coverage (τ)	0.2	0.2	0.2	0.2	0.2	0.2
+ Learning rate	0.01	0.01	0.01	0.01	0.01	0.01
+ λ_{lens}	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
+α_{KD}	0.9	0.9	0.9	0.9	0.9	0.9
+ T_{KD}	10	10	10	10	10	10
+hidden neurons	10	10	10	10	10	10
+λ_s	32	32	32	32	32	32
+ Temperature E-Lens (T_{lens})	0.7	0.7	0.7	0.7	0.7	0.7
CUB-200 (VIT)						
+ Batch size	16	16	16	16	16	16
+ Coverage (τ)	0.2	0.2	0.2	0.2	0.2	0.2
+ Learning rate	0.01	0.01	0.01	0.01	0.01	0.01
+ λ_{lens}	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
+α_{KD}	0.99	0.99	0.99	0.99	0.99	0.99
+ T_{KD}	10	10	10	10	10	10
+hidden neurons	10	10	10	10	10	10
+λ_s	32	32	32	32	32	32
+ Temperature E-Lens (T_{lens})	6.0	6.0	6.0	6.0	6.0	6.0

expert6 is the highest number of concepts utilized by any expert for CUB-200. Similarly, for Awa2, the highest number concept used by an expert is 8 for the VIT-based backbone compared to 80 for the ResNet-101-based backbone(Figure 10b).

A.8.2 Performance of experts and residual for ResNet-derived experts of Awa2 and CUB-200 datasets

Figure 11 shows the coverage (top row), performances (bottom row) of each expert and residual for the ResNet-101-derived experts of Awa2 and CUB-200 dataset respectively.

A.8.3 Comparison of performance with the Prototype-based interpretable models

Table 6 compares the performance of our model with the Prototype-based interpretable models (ProtoPNet Chen et al. (2019) and Prototree Nauta et al. (2021)). We list the following key differences between MoIE and the Prototype-based interpretable models:

1. Our method allows leveraging a blackbox and distilling it to any symbolic method (including ProtoTree), while a Prototype-based approach should be trained from scratch. Training from scratch can be a difficult optimization task, depending on the template or architecture of the interpretable method.
2. The samples routed to the last residuals can be viewed as a subset of data for which the template of the interpretable method is not appropriate. Neither Prototype nor ProtoTree offers such flexibility.
3. Using prototype approaches to fix undesirable properties such as shortcuts is not straightforward. We have shown that our method can easily be used for such applications.

Table 3: Hyperparameter setting of interpretable experts (g) trained on ResNet-101 (top) and VIT (bottom) blackboxes for the Awa2 dataset

Settings based on dataset	**Iteration 1**	**Iteration 2**	**Iteration 3**	**Iteration 4**	**Iteration 5**	**Iteration 6**
Awa2 (ResNet-101)						
+ Batch size	30	30	30	30	-	-
+ Coverage (τ)	0.4	0.35	0.35	0.25	-	-
+ Learning rate	0.001	0.001	0.001	0.001	-	-
+ λ_{lens}	0.0001	0.0001	0.0001	0.0001	-	-
+α_{KD}	0.9	0.9	0.9	0.9	-	-
+ T_{KD}	10	10	10	10	-	-
+hidden neurons	10	10	10	10	-	-
+λ_s	32	32	32	32	-	-
+ Temperature E-Lens (T_{lens})	0.7	0.7	0.7	0.7	-	-
Awa2 (VIT)						
+ Batch size	30	30	30	30	30	30
+ Coverage (τ)	0.2	0.2	0.2	0.2	0.2	0.2
+ Learning rate	0.01	0.01	0.01	0.01	0.01	0.01
+ λ_{lens}	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
+α_{KD}	0.99	0.99	0.99	0.99	0.99	0.99
+ T_{KD}	10	10	10	10	10	10
+hidden neurons	10	10	10	10	10	10
+λ_s	32	32	32	32	32	32
+ Temperature E-Lens (T_{lens})	6.0	6.0	6.0	6.0	6.0	6.0

Table 4: Hyperparameter setting of interpretable experts (g) for the diseases - Effusion (top) and Cardiomegaly (bottom) in the dataset HAM10000

Settings based on dataset	**Iteration 1**	**Iteration 2**	**Iteration 3**	**Iteration 4**	**Iteration 5**
HAM10000 (Inception-V3)					
+ Batch size	32	32	32	32	32
+ Coverage (τ)	0.4	0.2	0.2	0.2	0.2
+ Learning rate	0.01	0.01	0.01	0.01	0.01
+ λ_{lens}	0.0001	0.0001	0.0001	0.0001	0.0001
+α_{KD}	0.9	0.9	0.9	0.9	0.9
+ T_{KD}	10	10	10	10	10
+hidden neurons	10	10	10	10	10
+λ_s	64	64	64	64	64
+ Temperature E-Lens (T_{lens})	0.7	0.7	0.7	0.7	0.7

A.8.4 PERFORMANCE OF EXPERTS AND RESIDUAL FOR MEDICAL IMAGING DATASETS

Figure 12 shows the coverage (top row), auroc (middle row) and accuracy scores (bottom row) of each expert and residual for all the medical imaging datasets (HAM10000, Effusion of MIMIC-CXR and Cardiomegaly of MIMIC-CXR).

Table 5: Hyperparameter setting of interpretable experts (g) for the dataset MIMIC-CXR

Settings based on dataset	Iteration 1	Iteration 2	Iteration 3
Effusion-MIMIC-CXR (DenseNet-121)			
+ Batch size	64	64	64
+ Coverage (τ)	0.5	0.2	0.1
+ Learning rate	0.01	0.01	0.01
+ λ_{lens}	0.0001	0.0001	0.0001
+α_{KD}	0.99	0.99	0.99
+ T_{KD}	20	20	20
+hidden neurons	20, 20	20, 20	20, 20
+λ_s	96	128	256
+ Temperature			
E-Lens (T_{lens})	7.6	7.6	7.6
Cardiomegaly-MIMIC-CXR (DenseNet-121)			
+ Batch size	64	64	64
+ Coverage (τ)	0.5	0.15	0.1
+ Learning rate	0.01	0.01	0.01
+ λ_{lens}	0.0001	0.0001	0.0001
+α_{KD}	0.99	0.99	0.99
+ T_{KD}	20	20	20
+hidden neurons	20, 20	20, 20	20, 20
+λ_s	1024	64	256
+ Temperature			
E-Lens (T_{lens})	0.7	0.7	0.7

Table 6: Comparison of performance between MoIE and Prototype-based Model.

Method	Top-1 Accuracy (%)
ProtoPNet (Chen et al. (2019))	79.2
ProtoTree h=9 (Nauta et al. (2021))	82.2
MoIE (ours, ResNet Backbone)	88.64
MoIE (ours, VIT Backbone)	91.30

A.8.5 Results of MIMIC-CXR dataset

Figures 13 and 14 reveal the instances of local explanations for "Effusion" and "Cardiomegaly" respectively in MIMIC-CXR dataset.

A.8.6 Results of Awa2 dataset

Figure 15, shows the various local explanations for different species of animals in the Awa2 dataset. For brevity, we choose a maximum of 4 images per class in this figure. If an expert only includes one sample, we only show the image of that sample in this figure. For example, expert4 relies on the *water* concept to predict a "Beaver", whereas expert1 uses several other concepts such as *gray, nocturnal, muscle*. Figures 16 and 17 display the average number of concepts required to predict an animal species correctly in the Awa2 dataset for ResNet-101 and VIT as backbones, respectively. Specifically, the average number of concepts for class j = $\frac{\sum \text{all concepts for the samples belong to class } j}{\text{\# samples of class } j}$. We can see that for ResNet-101, on average, 80 concepts are required to explain a sample correctly for the class "Weasel" (Expert1 in Figure 16 a). However, for VIT, only three concepts are needed to explain a sample correctly for "Weasel" (Expert 6 in Figure 17 f). Also from both of these figures 16 and 17, we can see that different experts require different number concepts to explain same class.

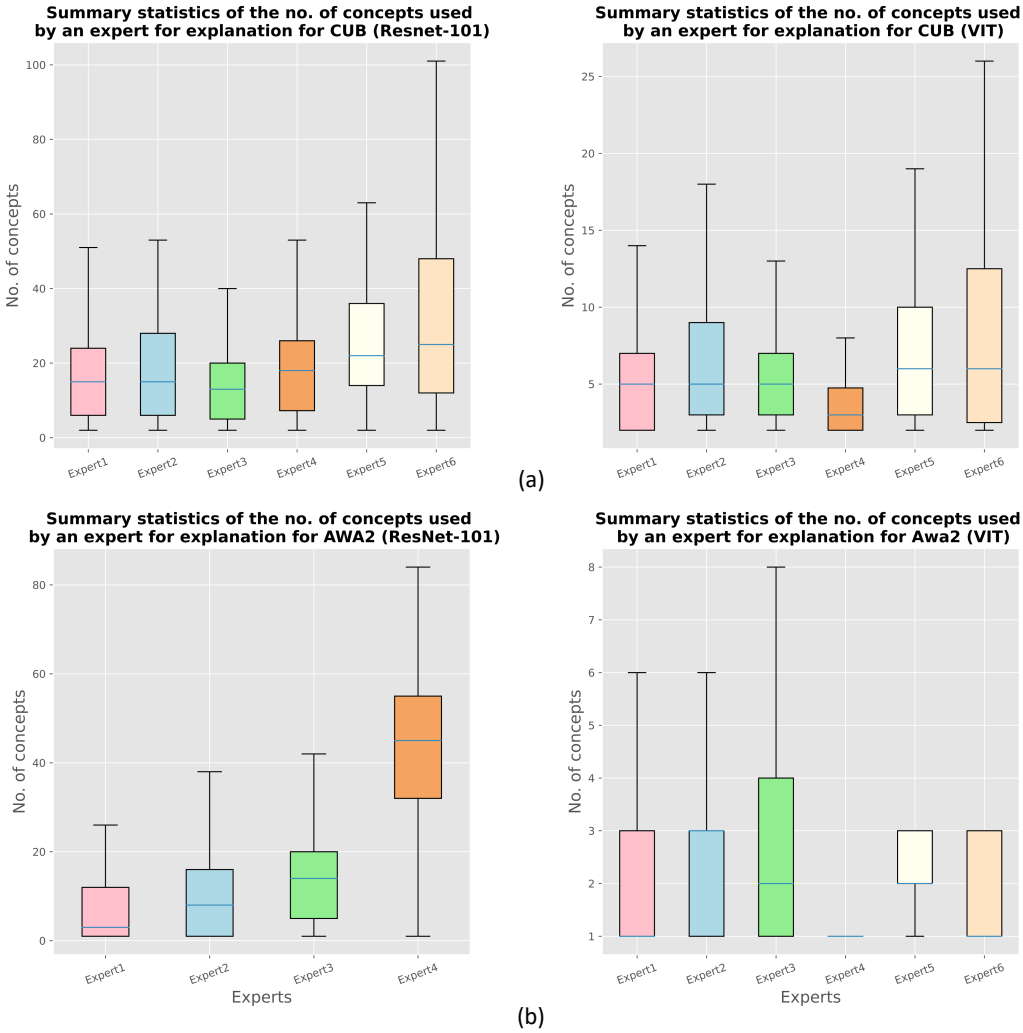

Figure 10: Comparison of summary statistics of the number of concepts utilized by various experts of datasets (a) CUB -200(top row) and (b) Awa2 (bottom row).. In general, we can see that experts carving out the explanations from VIT often uses less number of concepts.

For example figures 17 (e) and (f) reveal that experts 5 and 6 require 4 and 30 concepts on average to explain "Wolf" correctly.

A.8.7 More results of HAM1000 dataset

Figures 18 and 19 displays all unique individual FOL explanations by various experts to predict the skin lesions as "Malignant" and "Benign" correctly. In this figure, we observe that expert1 relies solely on the concept *Blue_Whitish_Veil(BWV)* to classify a skin lesion as "Benign", whereas expert3 relies on five different sets (one set consists of only *Is_Female, Regression_Structures etc.*, and another consists of *Is_Female, Irregular_Streaks etc.*). This result substantiates our hypothesis that different experts rely on different concepts for different diseases unlike the baselines in figures 18a and 19a detecting a skin lesion as "Malignant" and "Benign" respectively..

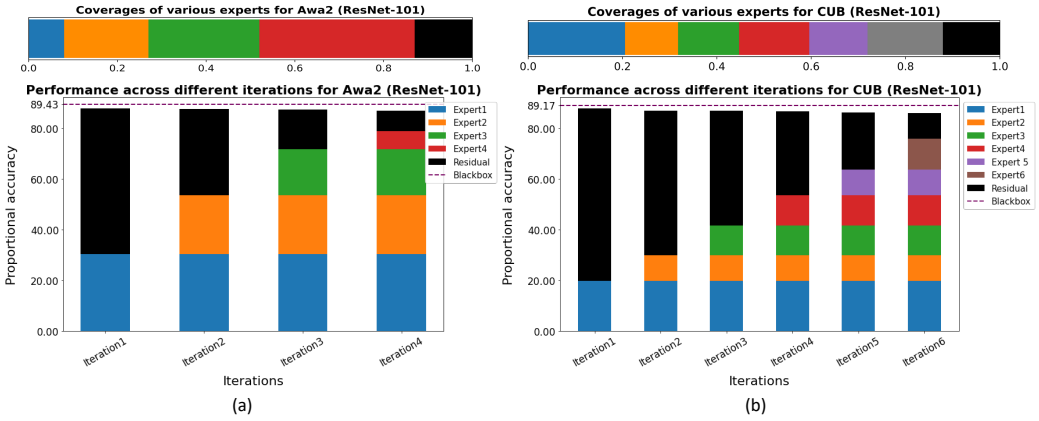

Figure 11: Coverage and performance of each expert and residual for the ResNet-101-derived experts of - (a) Awa2 and (b) CUB-200.

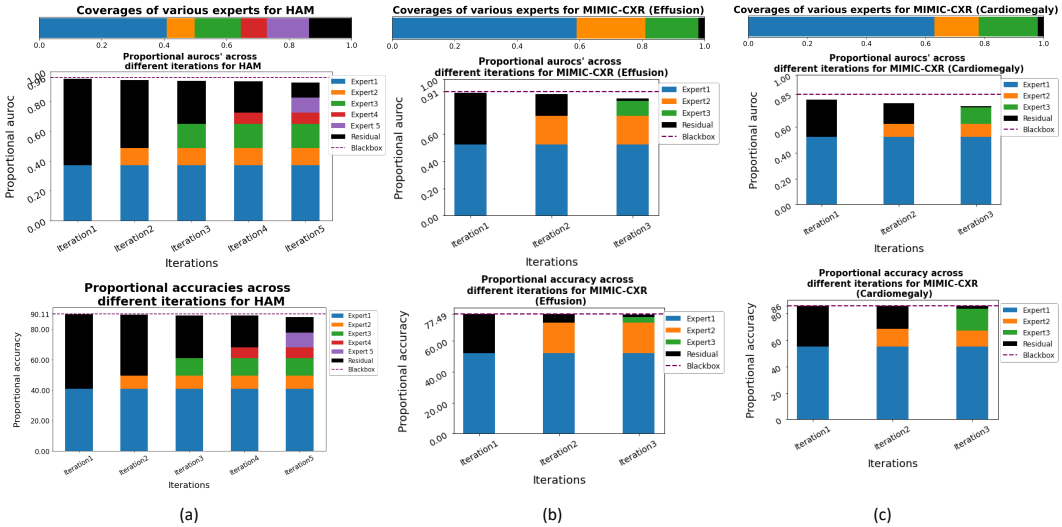

Figure 12: Coverage and performance of each expert and residual for all the medical imaging datasets - (a) HAM10000 (b) Effusion of MIMIC-CXR and (c) Cardiomegaly of MIMIC-CXR

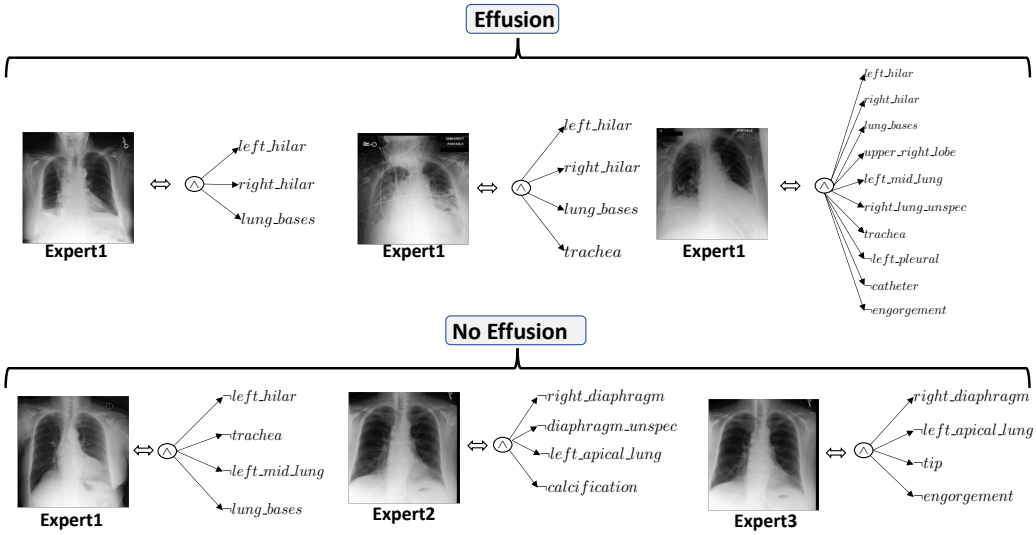

Figure 13: Sample local explanation for "Effusion" and "No Effusion" in MIMIC-CXR dataset.

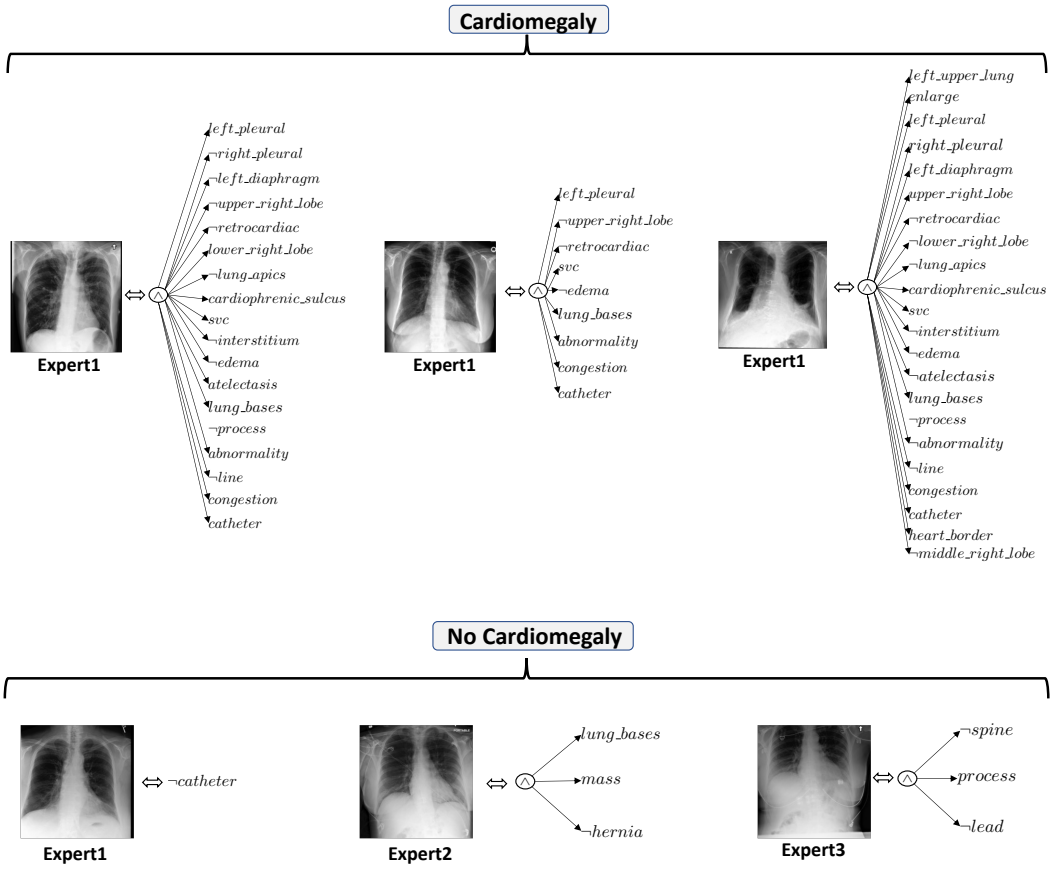

Figure 14: Sample local explanation for "Cardiomegaly" and "No Cardiomegaly" in MIMIC-CXR dataset.

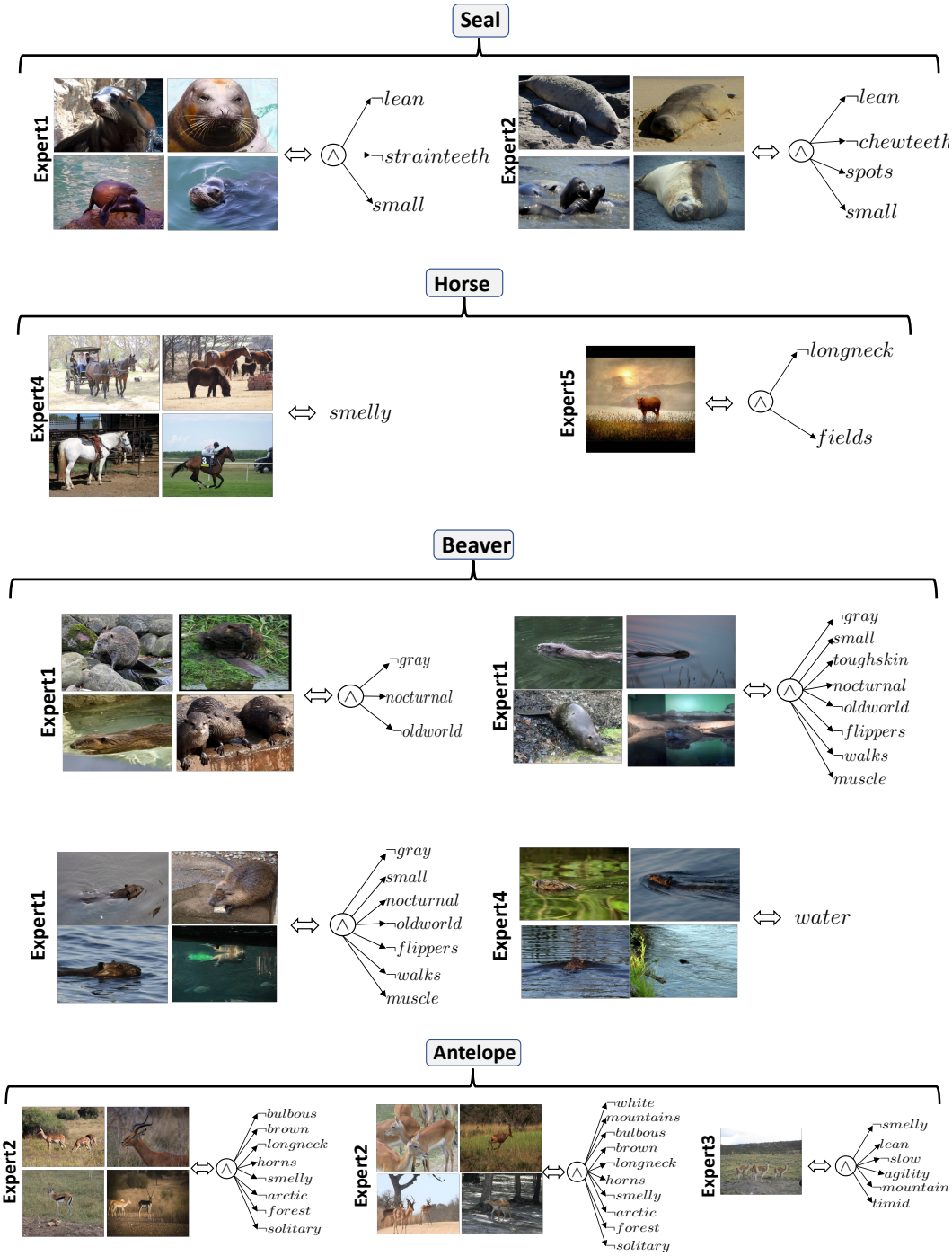

Figure 15: Examples of "Local explanations" from the different experts capture the variability of explanations for different samples for Awa2 for identifying an animal species as "Seal", "Horse" and "Beaver" respectively.

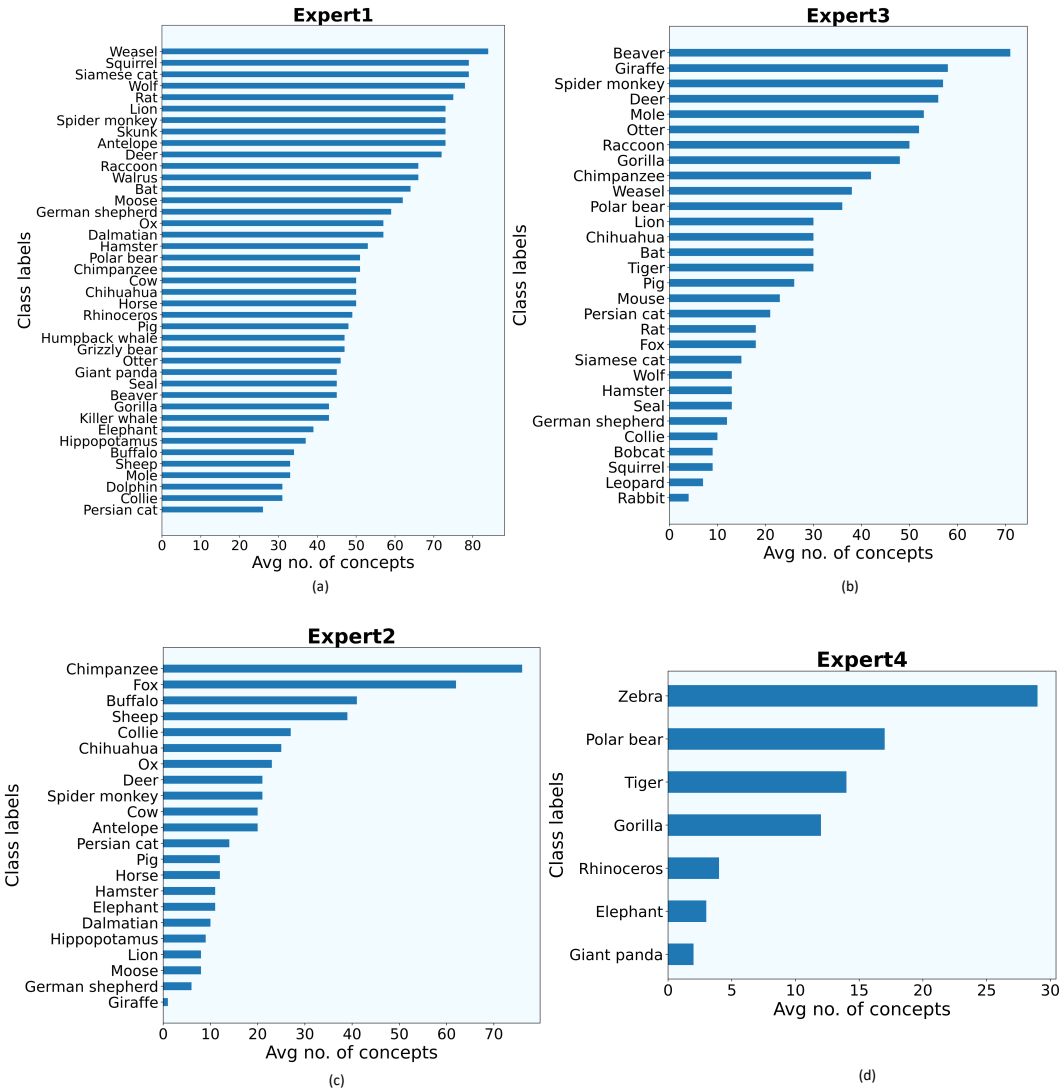

Figure 16: Class labels (Animal species) vs avg concepts using ResNet-101 as backbone for Awa2. Each bar in this plot indicates the average number concepts required to explain each sample of that animal species correctly.

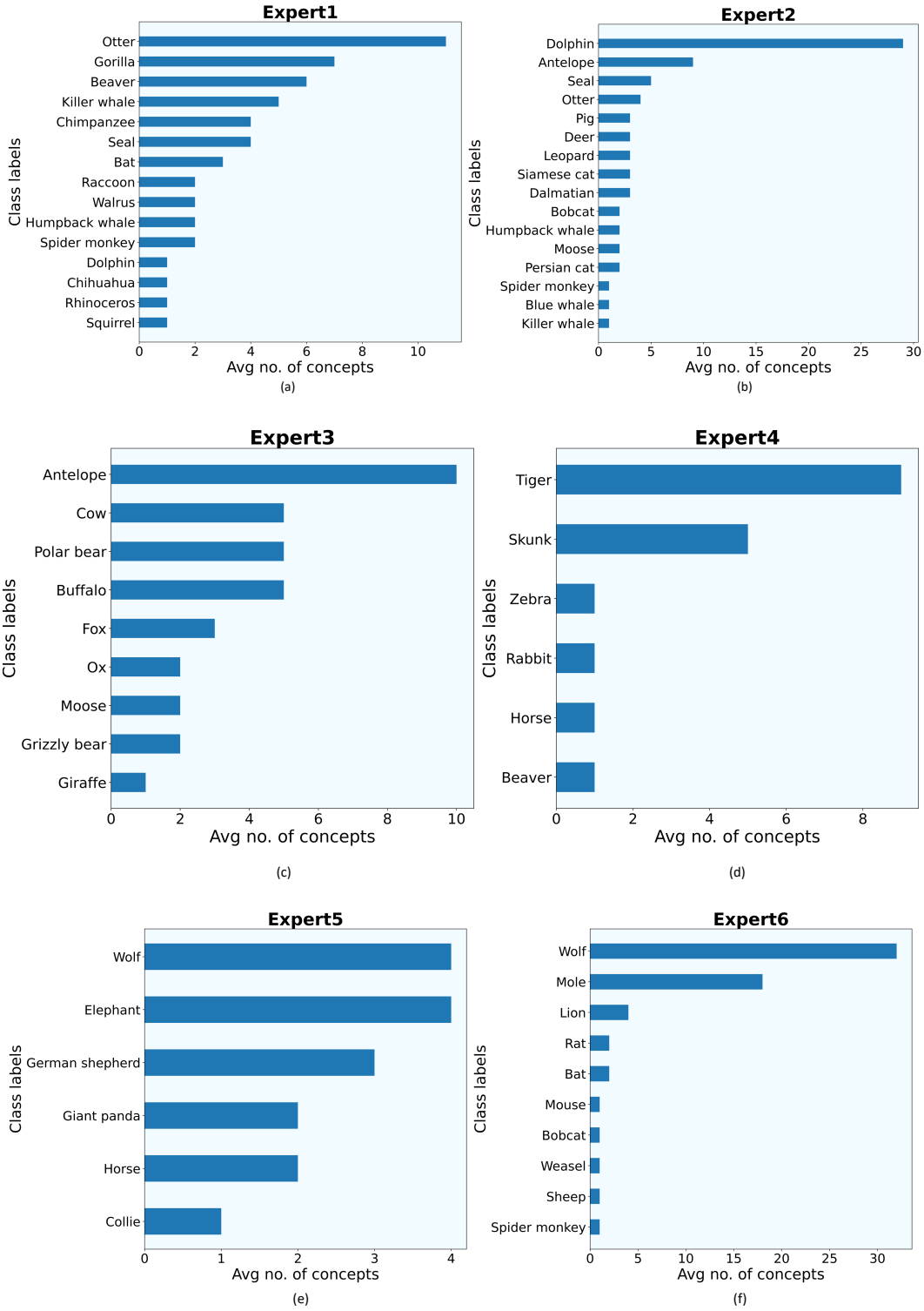

Figure 17: Class labels (Animal species) vs avg concepts using VIT as backbone for Awa2. Each bar in this plot indicates the average number concepts required to explain each sample of that animal species correctly.

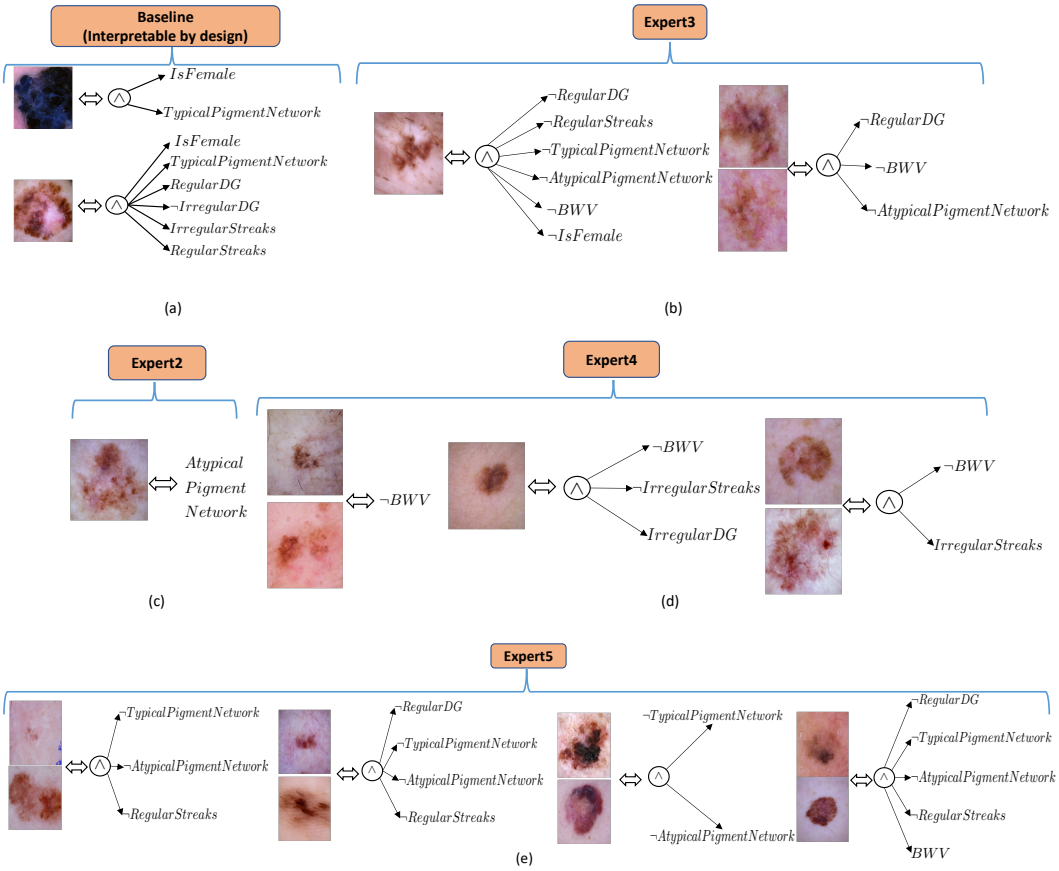

Figure 18: Local explanations from (a) the baseline and (b-e) different experts capture the variability of explanations for different samples for HAM10000 dataset for identifying a skin lesion as "Malignant".

A.8.8 Sample images covered by each VIT-derived expert and the final residual of CUB-200

Figure 20 compares different sample images covered by different VIT-derived experts and the final residuals of CUB-200. Figure 21 shows more instances, covered by the VIT-derived final residual of CUB-200. Table 7 compares the performance of the final residual with that of the blackbox (f^0). The second column of the table shows the performance of the blackbox (f^0) on the samples covered by the final residua. The third column shows the performance of the blackbox (f^0) on all the samples in the test set. Clearly, this table shows that the performance of the blackbox (f^0) drops substantially for the samples covered by the final residual. For example, for HAM10000, the overall performance of the blackbox (f^0) is 92.15%. However, on the samples covered by the final residual, the performance of the blackbox (f^0) drops to 67.89%. This experiment demonstrates that the final residual is left with relatively "harder" samples to explain.

A.9 Validity of the generated explanations

To ensure the validity of the FOL explanations, we intervene on the concepts in the derived FOL and set the values of those concepts to zero. For example, *wing_shape_roundedwings, back_pattern_multicolored, bill_color_grey* and *head_pattern_plain* concepts show up in the FOL explanation of expert1 for the class **Baltimore Oriole** as per figure 28. For random intervention, we set the values of these concepts to zero and keep the values of other concepts unchanged. Then we pass the complete intervened concept vector as input to the corresponding expert and compute the accuracy. We discover that MoIE is highly sensitive to a random intervention on these concepts, and the performance of MoIE drops significantly. For example, as a result of the random interven-

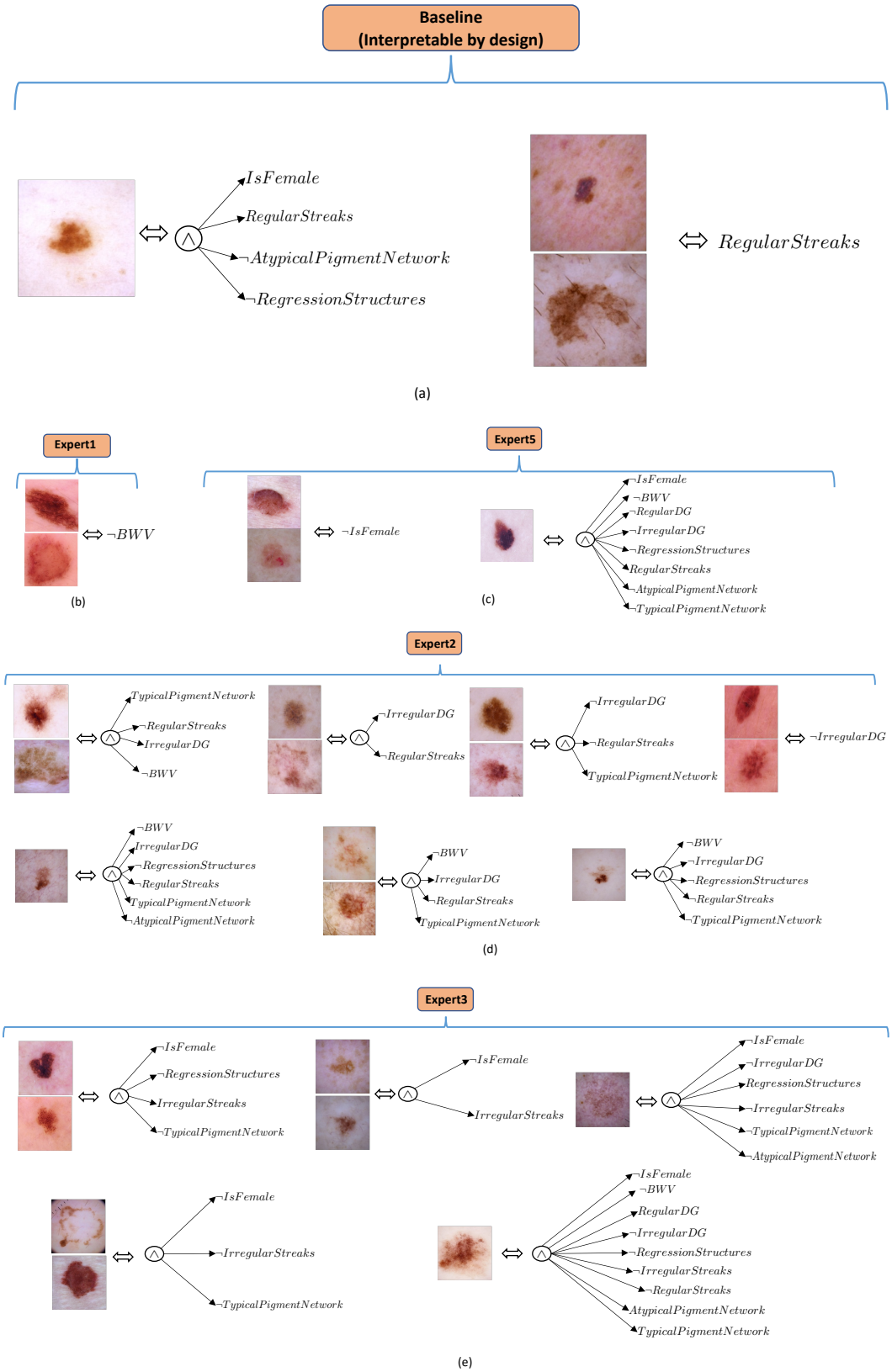

Figure 19: "Local explanations" from (a) the baseline and (b-e) different sparse experts capture the variability of explanations for different samples for HAM10000 dataset for identifying a skin lesion as "Benign".

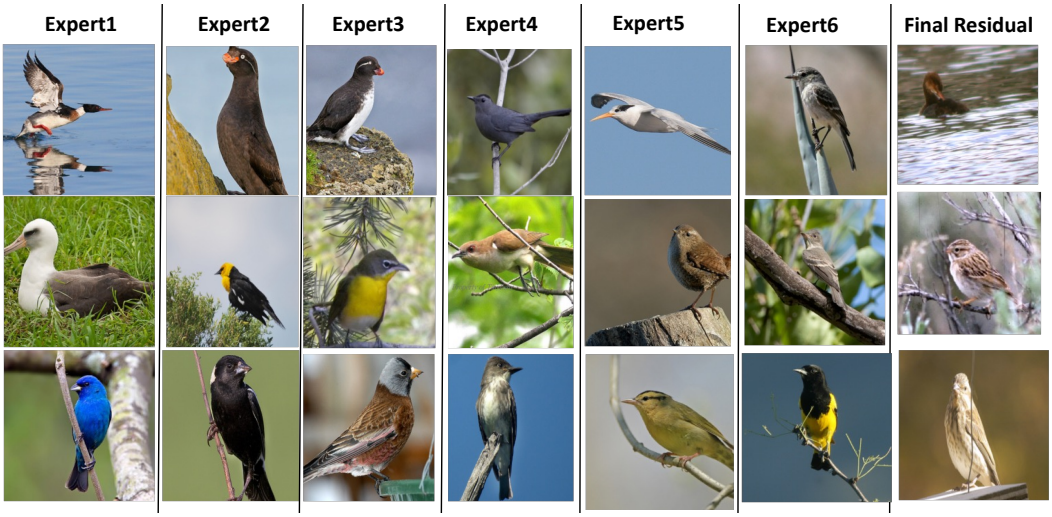

Figure 20: Sample images covered by each VIT-derived expert and the final residual of CUB-200.

Figure 21: More images covered by the VIT-derived final residual of CUB-200.

tion, for CUB-200 VIT-derived MoIE, the performance of MoIE deteriorates from 91.30 to 60.13 % (a 34.09 % drop). We perform the identical experiment for the baseline Koh et al. (2020). For CUB-200 VIT-based baseline model, the performance of the baseline degrades from 85.20 to 65.02 % (a 20.18 % drop). As a result, we infer that MoIE generates more concrete explanations than the baseline, as the drop in accuracy for the baseline is lower than that of MoIE. Table 8 demonstrates this experiment.

A.9.1 More results of CUB dataset

Figures 22, 23 and 24 display the average number of concepts required to predict a bird species correctly in the Cub-200 dataset for all the experts of VIT as backbones. Also, Figures 25, 27 and 27

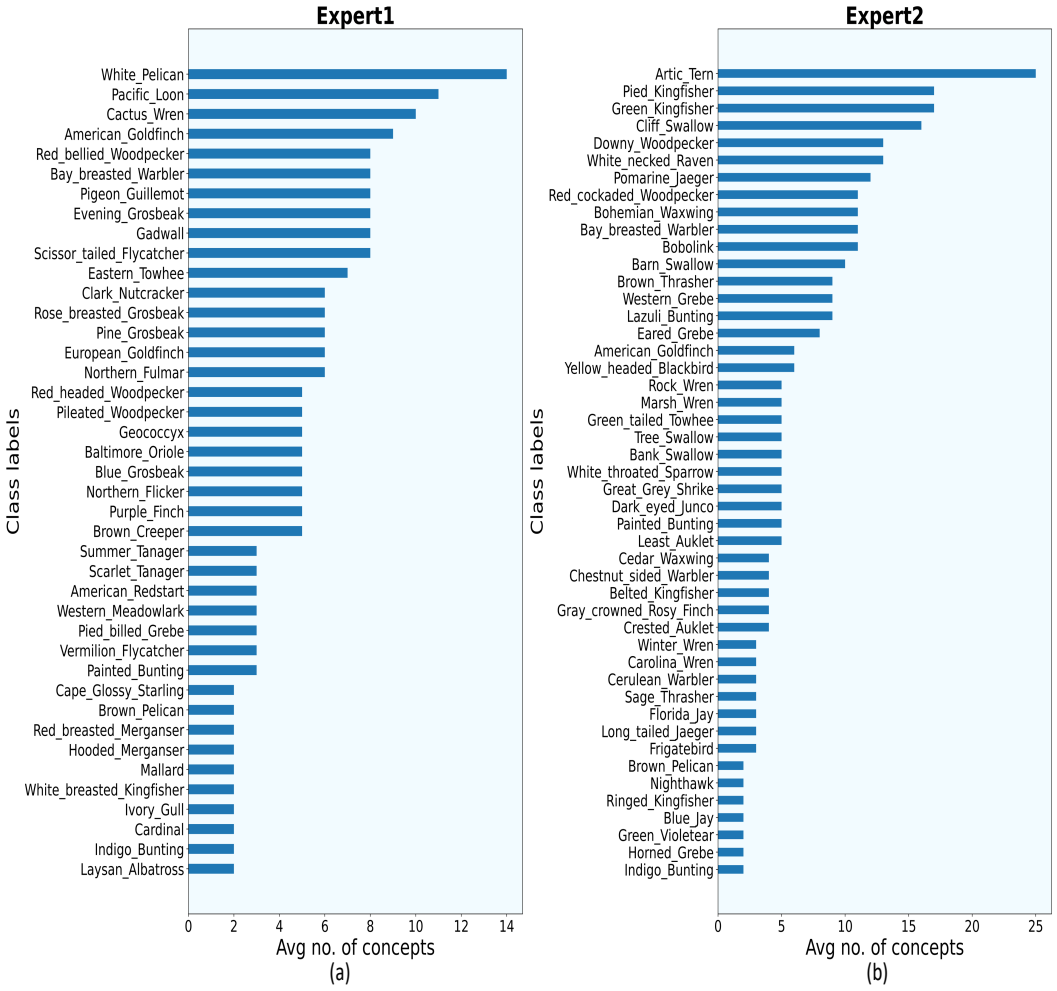

Figure 22: Class labels (Bird species) vs avg concepts using VIT as backbone for CUB-200 by (a) Expert1 (b) Expert2. Each bar in this plot indicates the average number concepts required to explain each sample of that bird species correctly.

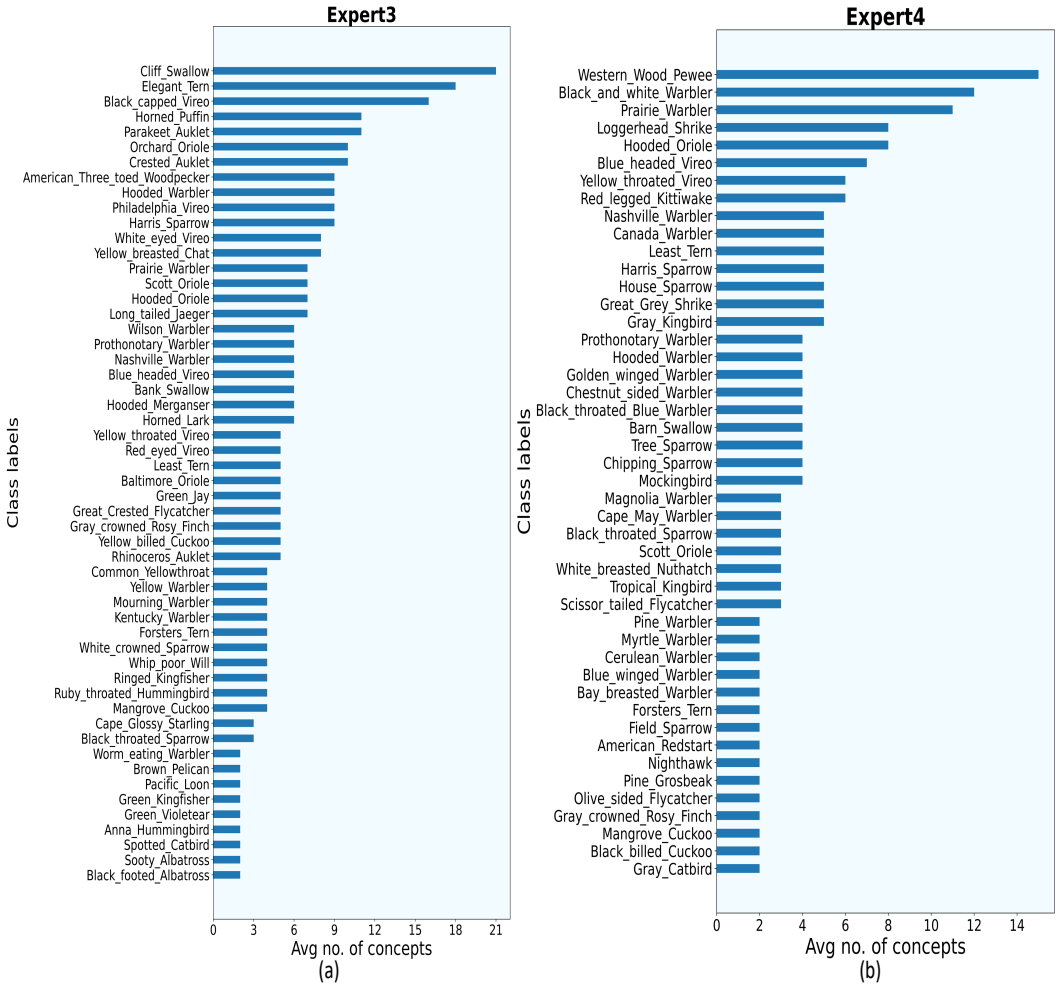

Figure 23: Class labels (Bird species) vs avg concepts using VIT as backbone for CUB-200 by (a) Expert3 (b) Expert4. Each bar in this plot indicates the average number concepts required to explain each sample of that bird species correctly.

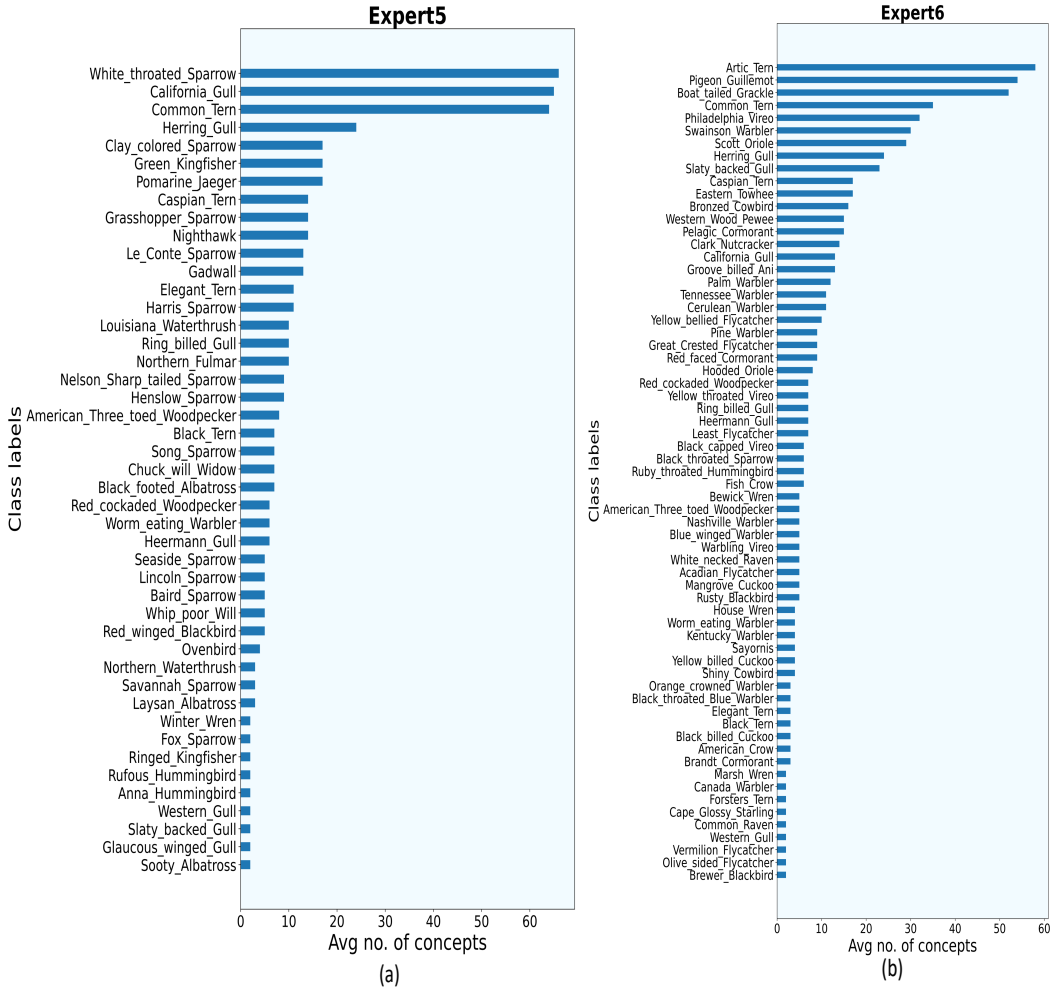

Figure 24: Class labels (Bird species) vs avg concepts using VIT as backbone for CUB-200 by (a) Expert5 (b) Expert6. Each bar in this plot indicates the average number concepts required to explain each sample of that bird species correctly.

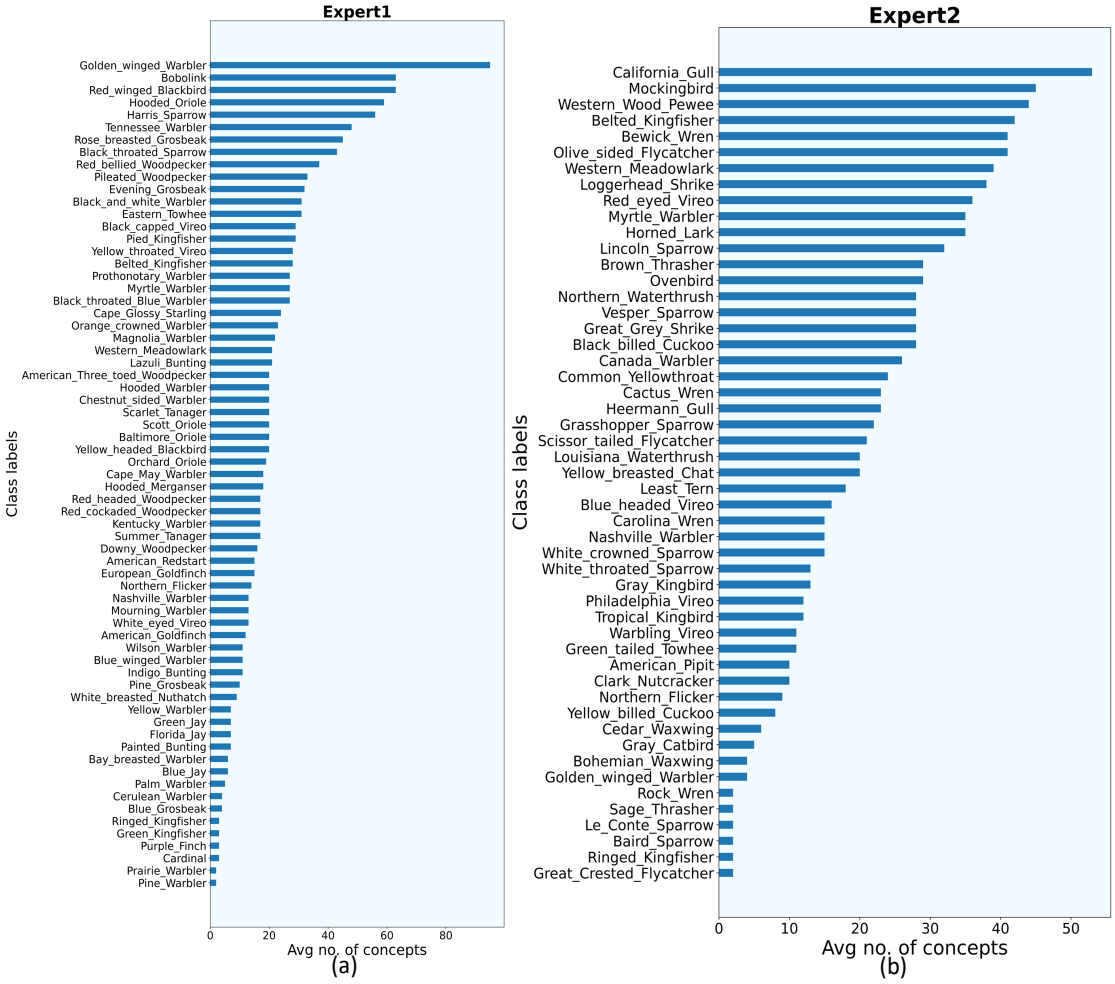

Figure 25: Class labels (Bird species) vs avg concepts using ResNet-101 as backbone for CUB-200 by (a) Expert1 (b) Expert2. Each bar in this plot indicates the average number concepts required to explain each sample of that bird species correctly.

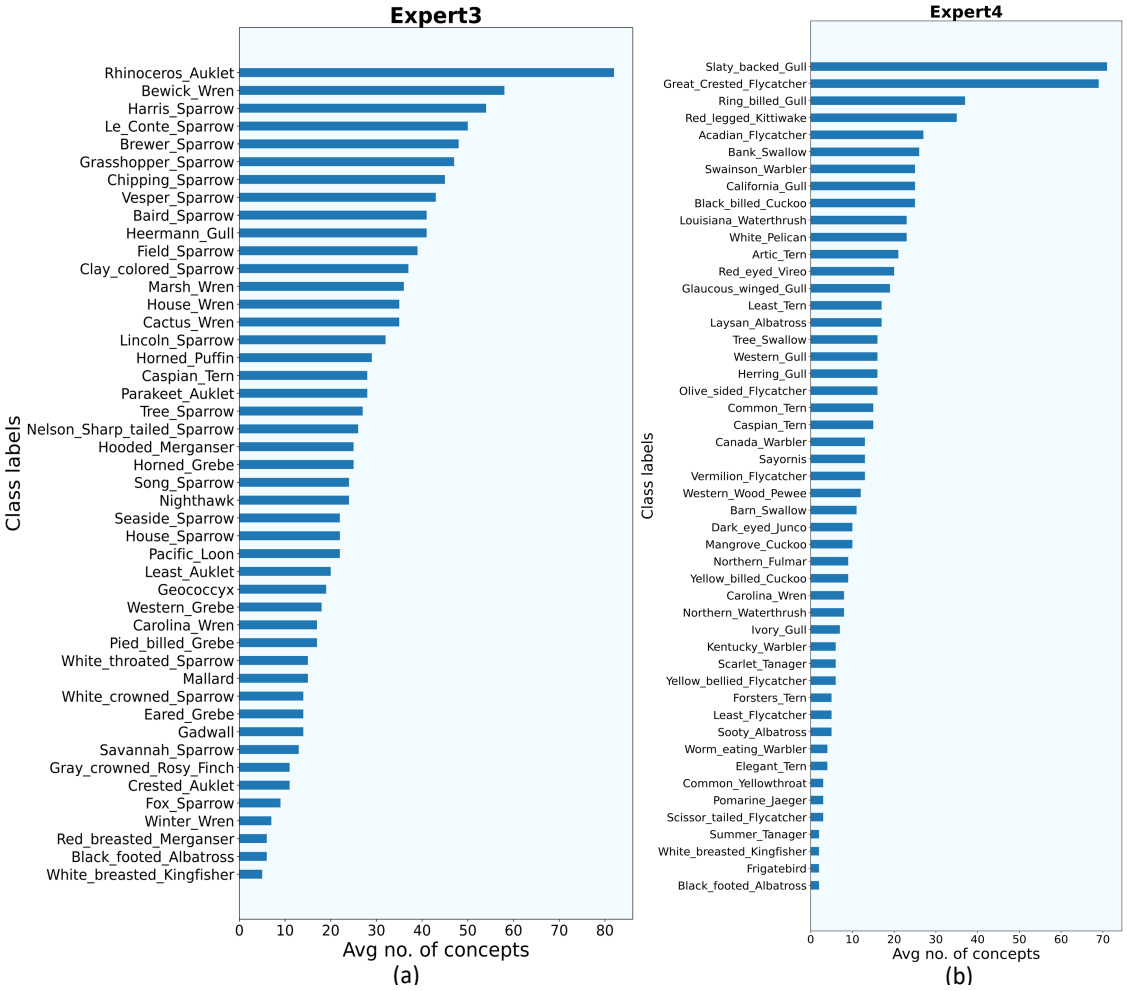

Figure 26: Class labels (Bird species) vs avg concepts using ResNet-101 as backbone for CUB-200 by (a) Expert3 (b) Expert4. Each bar in this plot indicates the average number concepts required to explain each sample of that bird species correctly.

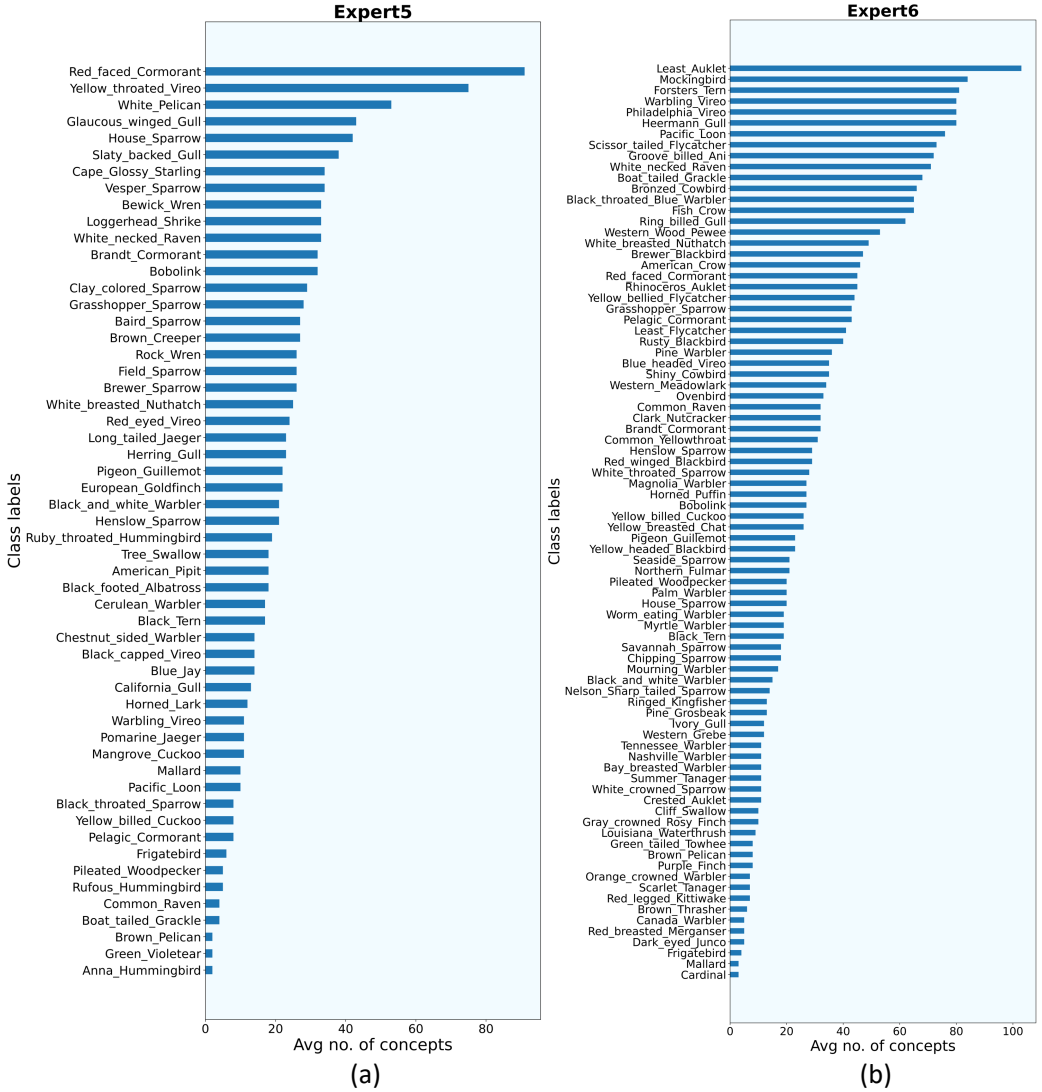

Figure 27: Class labels (Bird species) vs avg concepts using ResNet-101 as backbone for CUB-200 by (a) Expert5 (b) Expert6. Each bar in this plot indicates the average number concepts required to explain each sample of that bird species correctly.

Table 7: Comparison of performance (Accuracy, %) of the final residual and the blackbox (f^0) on held-out test set. The 2^{nd} column depcits the performance of the initial blackbox (f^0) on the samples of the test set. To compare fairly, we dispatch the samples, covered by the final residual through the blackbox (f^0) and compare the performance of the blackbox on these samples with the final residual (3^{rd} column). We observe that the performance of blackbox on the samples covered by the final residual is lower than that of the blackbox on all the samples for all the dataset.

Dataset (Architecture)	**On all samples (%)**	**On selected samples covered by the final residual (%)** (residual, blackbox)
Cub-200 (ResNet-101)	88.64	(82.52, 85.41)
Cub-200 (VIT)	91.30	(81.01, 83.01)
Awa2 (ResNet-101)	91.02	(77.88, 79.11)
Awa2 (VIT)	98.53	(92.56, 93.56)
HAM10000 (Inception)	92.15	(67.89, 62.89)
Effusion from MIMIC-CXR (DenseNet-121)	78.34	(35.71, 37.06)
Cardiomegaly from MIMIC-CXR (DenseNet-121)	84.89	(48.71, 51.06)

Table 8: Explanation validity for MoIE / baseline. The 2^{nd} column depicts the accuracy of MoIE / baseline using the discovered concepts in the FOL per sample. The 3^{rd} column depicts the accuracy of MoIE / baseline using the intervened concepts in the FOL per sample. The 4^{th} column shows the drop in accuracy. The more drop in accuracy illustrates the model to be more sensitive to random intervention of the derived concepts.

Dataset (Architecture)	**Accuracy using correct concepts (%)**	**Accuracy using intervened concepts (%)**	**Drop(%) ↓**
MoIE (ours)			
+ Cub-200 (ResNet-101)	88.64	54.33	34.31
+ Cub-200 (VIT)	91.30	60.17	31.14
+ Awa2 (ResNet-101)	91.02	53.23	37.79
+ Awa2 (VIT)	98.53	90.19	8.34
+ HAM10000 (Inception)	92.15	86.72	5.43
+ Effusion - MIMIC-CXR (DenseNet-121)	78.34	72.32	6.02
+ Cardiomegaly - MIMIC-CXR (DenseNet-121)	84.89	82.17	2.72
Baseline (interpretable by design Koh et al. (2020))			
+ Cub-200 (ResNet-101)	74.80	53.16	21.64
+ Cub-200 (VIT)	85.20	65.02	20.18
+ Awa2 (ResNet-101)	90.05	88.58	1.47
+ Awa2 (VIT)	95.80	95.25	0.55
+ HAM10000 (Inception)	84.97	82.44	2.53
+ Effusion - MIMIC-CXR (DenseNet-121)	78.11	77.85	0.26
+ Cardiomegaly - MIMIC-CXR (DenseNet-121)	84.30	83.98	0.32

display the same for the ResNet-101 based counterparts. As mentioned before, the average number of concepts for class j = $\frac{\sum \text{all concepts for the samples belong to class } j}{\text{\# samples of class } j}$. We can see that for ResNet-101, on average 80 concepts are required to explain a sample correctly for the class "Rhinoceros_Auklet" (expert3 in Figure 27 a). However, for VIT, only 6 concepts are needed to explain a sample correctly "Rhinoceros_Auklet" (expert3 in Figure 27 a). From both of these figures, we can see that different experts require a different number of concepts to explain the same class. For example, figures 22 (b) and 24 (b) reveal that experts 2 and 6 require 25 and 58 concepts on average to explain "Artic_Tern"

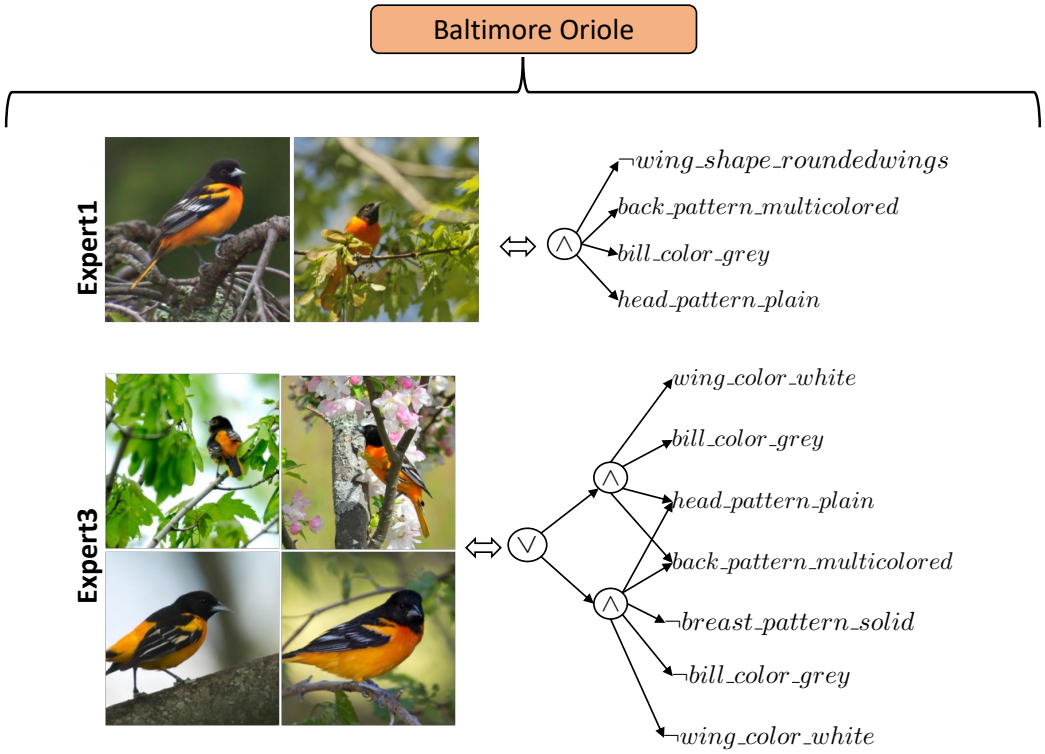

Figure 28: Generation of class-level explanations from VIT as a blackbox by combining the local explanations of "Baltimore Oriole" by expert1(top row) and expert3 (bottom row).

correctly respectively. Figures 28, 29, 30 show more results on global explanations of CUB-200 dataset.

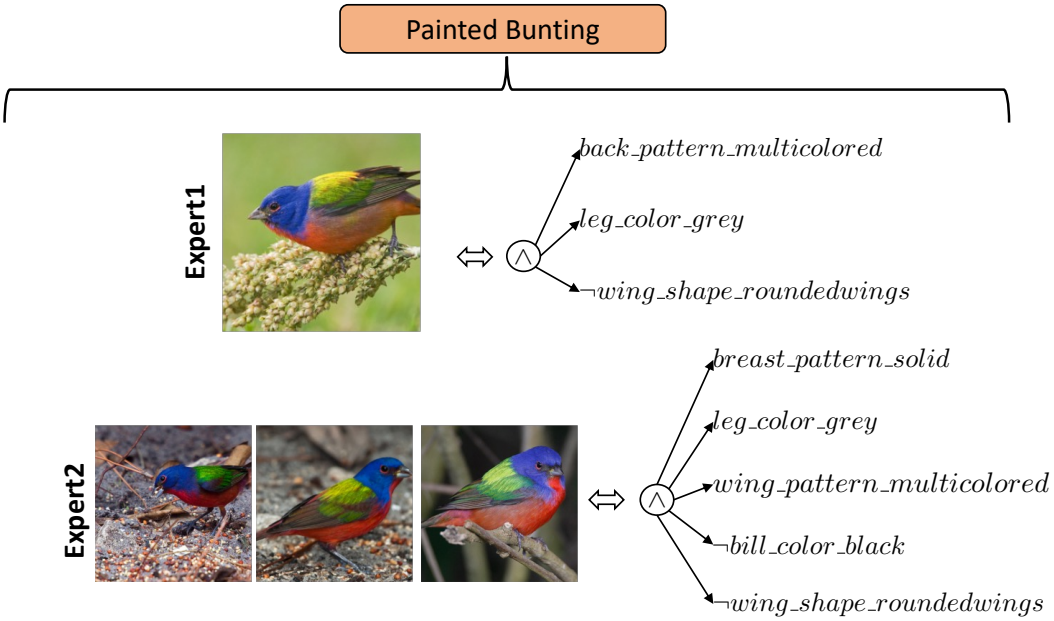

Figure 29: Generation of class-level explanations from VIT as a blackbox by combining the local explanations of "Painted Bunting" by expert1(top row) and expert2 (bottom row). Note that all the samples have same local explanation, so their local and global explanation is same.

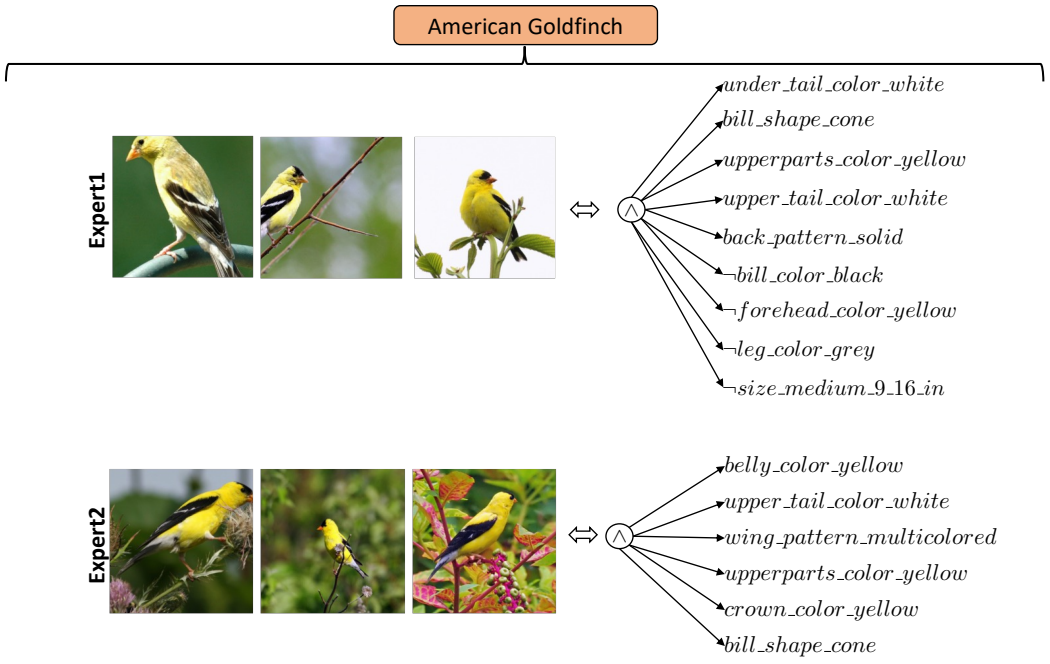

Figure 30: Generation of class-level explanations from VIT as a blackbox by combining the local explanations of "American Goldfinch" by expert1(top row) and expert2 (bottom row). Note that all the samples have same local explanation, so their local and global explanation is same.