Dividing and Conquering a BlackBox to a Mixture of Interpretable Models: Route, Interpret, Repeat

Shantanu Ghosh [1] **Ke Yu** [2] **Forough Arabshahi** [3] **Kayhan Batmanghelich** [1]

Abstract

ML model design either starts with an interpretable model or a Blackbox and explains it post hoc. Blackbox models are flexible but difficult to explain, while interpretable models are inherently explainable. Yet, interpretable models require extensive ML knowledge and tend to be less flexible and underperforming than their Blackbox variants. This paper aims to blur the distinction between a post hoc explanation of a Blackbox and constructing interpretable models. Beginning with a Blackbox, we iteratively *carve out* a mixture of interpretable experts (MoIE) and a *residual network*. Each interpretable model specializes in a subset of samples and explains them using First Order Logic (FOL), providing basic reasoning on concepts from the Blackbox. We route the remaining samples through a flexible residual. We repeat the method on the residual network until all the interpretable models explain the desired proportion of data. Our extensive experiments show that our *route, interpret, and repeat* approach (1) identifies a diverse set of instance-specific concepts with high concept completeness via MoIE without compromising in performance, (2) identifies the relatively "harder" samples to explain via residuals, (3) outperforms the interpretable by-design models by significant margins during test-time interventions, and (4) fixes the shortcut learned by the original Blackbox. The code for MoIE is publicly available at: `https://github.com/batmanlab/ICML-2023-Route-interpret-repeat`.

1. Introduction

Model explainability is essential in high-stakes applications of AI, *e.g.,* healthcare. While Blackbox models (*e.g.,* Deep Learning) offer flexibility and modular design, post hoc explanation is prone to confirmation bias (Wan et al., 2022), lack of fidelity to the original model (Adebayo et al., 2018), and insufficient mechanistic explanation of the decision-making process (Rudin, 2019). Interpretable-by-design models overcome those issues but tend to be less flexible than Blackbox models and demand substantial expertise to design. Using a post hoc explanation or adopting an inherently interpretable model is a mutually exclusive decision to be made at the initial phase of AI model design. This paper blurs the line on that dichotomous model design.

The literature on post hoc explanations is extensive. This includes model attributions ((Simonyan et al., 2013; Selvaraju et al., 2017)), counterfactual approaches (Abid et al., 2021; Singla et al., 2019), and distillation methods (Alharbi et al., 2021; Cheng et al., 2020). Those methods either identify key input features that contribute the most to the network's output (Shrikumar et al., 2016), generate input perturbation to flip the network's output (Samek et al., 2016; Montavon et al., 2018), or estimate simpler functions to approximate the network output locally. Post hoc methods preserve the flexibility and performance of the Blackbox but suffer from a lack of fidelity and mechanistic explanation of the network output (Rudin, 2019). Without a mechanistic explanation, recourse to a model's undesirable behavior is unclear. Interpretable models are alternative designs to the Blackbox without many such drawbacks. For example, modern interpretable methods highlight human understandable *concepts* that contribute to the downstream prediction.

Several families of interpretable models exist for a long time, such as the rule-based approach and generalized additive models (Hastie & Tibshirani, 1987; Letham et al., 2015; Breiman et al., 1984). They primarily focus on tabular data. Such models for high-dimensional data (*e.g.,* images) primarily rely on projecting to a lower dimensional human understandable *concept* or *symbolic* space (Koh et al., 2020) and predicting the output with an interpretable classifier. Despite their utility, the current State-Of-The-Art (SOTA) are limited in design; for example, they do not model the

[1] Department of Electrical and Computer Engineering, Boston University, MA, USA [2] Intelligent Systems Program, University of Pittsburgh, PA, USA [3] MetaAI, MenloPark, CA, USA. Correspondence to: Shantanu Ghosh <shawn24@bu.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

interaction between the concepts except for a few exceptions (Ciravegna et al., 2021; Barbiero et al., 2022), offering limited reasoning capabilities and robustness. Furthermore, if a portion of the samples does not fit the template design of the interpretable model, they do not offer any flexibility, compromising performance.

Our contributions We propose an interpretable method, aiming to achieve the best of both worlds: not sacrificing Blackbox performance similar to post hoc explainability while still providing actionable interpretation. We hypothesize that a Blackbox encodes several interpretable models, each applicable to a different portion of data. Thus, a single interpretable model may be insufficient to explain all samples. We construct a hybrid neuro-symbolic model by progressively *carving out* a mixture of interpretable models and a *residual network* from the given Blackbox. We coin the term *expert* for each interpretable model, as they specialize over a subset of data. All the interpretable models are termed a "Mixture of Interpretable Experts" (MoIE). Our design identifies a subset of samples and *routes* them through the interpretable models to explain the samples with FOL, providing basic reasoning on concepts from the Blackbox. The remaining samples are routed through a flexible residual network. On the residual network, we repeat the method until MoIE explains the desired proportion of data. We quantify the sufficiency of the identified concepts to explain the Blackbox's prediction using the concept completeness score (Yeh et al., 2019). Using FOL for interpretable models offers recourse when undesirable behavior is detected in the model. We provide an example of fixing a shortcut learning by modifying the FOL. FOL can be used in human-model interaction (not explored in this paper). Our method is the divide-and-conquer approach, where the instances covered by the residual network need progressively more complicated interpretable models. Such insight can be used to inspect the data and the model further. Finally, our model allows *unexplainable* category of data, which is currently not allowed in the interpretable models.

2. Method

Notation: Assume we have a dataset $\{\mathcal{X}, \mathcal{Y}, \mathcal{C}\}$, where $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{C}$ are the input images, class labels, and human interpretable attributes, respectively. $f^0 : \mathcal{X} \rightarrow \mathcal{Y}$, is our pre-trained initial Blackbox model. We assume that f^0 is a composition $h^0 \circ \Phi$, where $\Phi : \mathcal{X} \rightarrow \mathbb{R}^l$ is the image embeddings and $h^0 : \mathbb{R}^l \rightarrow \mathcal{Y}$ is a transformation from the embeddings, Φ, to the class labels. We denote the learnable function $t : \mathbb{R}^l \rightarrow \mathcal{C}$, projecting the image embeddings to the concept space. The concept space is the space spanned by the attributes $\mathcal{C}$. Thus, function t outputs a scalar value representing a concept for each input image.

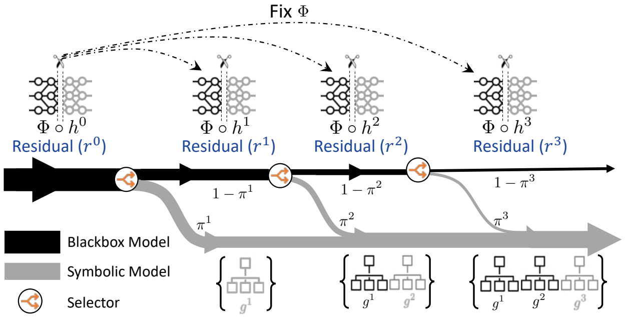


Figure 1. Schematic view of *route, interpret* and *repeat*. At iteration k, the selector *routes* each sample either towards the interpretable model g^k (to *interpret*) with probability $\pi^k(.)$ or the residual $r^k = f^{k-1} - g^k$ with probability $1 - \pi^k(.)$ (to *repeat* in the further iterations). f^{k-1} is the Blackbox of the $(k-1)^{th}$ iteration. g^k generates FOL-based explanations for the samples it covers. Otherwise, the selector routes through the next step until it either goes through a subsequent interpretable model or reaches the last residual. Components in black and grey indicate the fixed and trainable modules in our model, respectively.

Method Overview: Figure 1 summarizes our approach. We iteratively carve out an interpretable model from the given Blackbox. Each iteration yields an interpretable model (the downward grey paths in Figure 1) and a residual (the straightforward black paths in Figure 1). We start with the initial Blackbox f^0. At iteration k, we distill the Blackbox from the previous iteration f^{k-1} into a neuro-symbolic interpretable model, $g^k : \mathcal{C} \rightarrow \mathcal{Y}$. Our g is flexible enough to be any interpretable models (Yuksekgonul et al., 2022; Koh et al., 2020; Barbiero et al., 2022). The *residual* $r^k = f^{k-1} - g^k$ emphasizes the portion of f^{k-1} that g^kcannot explain. We then approximate r^k with $f^k = h^k \circ \Phi$. f^k will be the Blackbox for the subsequent iteration and be explained by the respective interpretable model. A learnable gating mechanism, denoted by $\pi^k : \mathcal{C} \rightarrow \{0, 1\}$ (shown as the *selector* in Figure 1) routes an input sample towards either g^k or r^k. The thickness of the lines in Figure 1 represents the samples covered by the interpretable models (grey line) and the residuals (black line). With every iteration, the cumulative coverage of the interpretable models increases, but the residual decreases. We name our method *route, interpret* and *repeat.*

2.1. Neuro-Symbolic Knowledge Distillation

Knowledge distillation in our method involves 3 parts: (1) a series of trainable selectors, *routing* each sample through the interpretable models and the residual networks, (2) a sequence of learnable neuro-symbolic interpretable models, each providing FOL explanations to *interpret* the Blackbox, and (3) *repeating* with Residuals for the samples that cannot be explained with their interpretable counterparts. We detail each component below.

2.1.1. The selector function

As the first step of our method, the selector π^k *routes* the j^{th} sample through the interpretable model g^k or residual r^k with probability $\pi^k(\boldsymbol{c_j})$ and $1 - \pi^k(\boldsymbol{c_j})$ respectively, where $k \in [0, K]$, with K being the number of iterations. We define the empirical coverage of the k^{th} iteration as $\zeta(\pi^k) = \frac{1}{m}\sum_{j=1}^{m}\pi^k(\boldsymbol{c_j})$, the empirical mean of the samples selected by the selector for the associated interpretable model g^k, with m being the total number of samples in the training set. Thus, the entire selective risk is:

$$\mathcal{R}^k(\pi^k, g^k) = \frac{\frac{1}{m}\sum_{j=1}^{m}\mathcal{L}^k_{(g^k,\pi^k)}\big(\boldsymbol{x_j}, \boldsymbol{c_j}\big)}{\zeta(\pi^k)}, \tag{1}$$

where $\mathcal{L}^k_{(g^k,\pi^k)}$ is the optimization loss used to learn g^k and π^k together, discussed in Section 2.1.2. For a given coverage of $\tau^k \in (0, 1]$, we solve the following optimization problem:

$$\begin{aligned} \theta^*_{s^k}, \theta^*_{g^k} &= \underset{\theta_{s^k}, \theta_{g^k}}{\arg\min}\, \mathcal{R}^k\Big(\pi^k(.;\theta_{s^k}), g^k(.;\theta_{g^k})\Big) \\ &\text{s.t.} \quad \zeta\big(\pi^k(.;\theta_{s^k})\big) \geq \tau^k, \end{aligned} \tag{2}$$

where $\theta^*_{s^k}, \theta^*_{g^k}$ are the optimal parameters at iteration k for the selector π^k and the interpretable model g^k respectively. In this work, πs' of different iterations are neural networks with sigmoid activation. At inference time, the selector routes the j^{th} sample with concept vector $\boldsymbol{c_j}$ to g^k if and only if $\pi^k(\boldsymbol{c}_j) \geq 0.5$ for $k \in [0, K]$.

2.1.2. Neuro-Symbolic interpretable models

In this stage, we design interpretable model g^k of k^{th} iteration to *interpret* the Blackbox f^{k-1} from the previous $(k-1)^{th}$ iteration by optimizing the following loss function:

$$\mathcal{L}^k_{(g^k,\pi^k)}(\boldsymbol{x_j}, \boldsymbol{c_j}) = \underbrace{\ell\Big(f^{k-1}(\boldsymbol{x_j}), g^k(\boldsymbol{c_j})\Big)\pi^k(c_j)}_{\substack{\text{trainable component}\\ \text{for current iteration } k}} \underbrace{\prod_{i=1}^{k-1}\big(1 - \pi^i(\boldsymbol{c_j})\big)}_{\substack{\text{fixed component trained}\\ \text{in the previous iterations}}}, \tag{3}$$

where the term $\pi^k(\boldsymbol{c_j})\prod_{i=1}^{k-1}\big(1 - \pi^i(\boldsymbol{c_j})\big)$ denotes the probability of j^{th} sample being routed through the interpretable model g^k. It is the probability of the sample going through the residuals for all the previous iterations from 1 through $k-1$ (*i.e.,* $\prod_{i=1}^{k-1}\big(1 - \pi^i(\boldsymbol{c_j})\big)$) times the probability of going through the interpretable model at iteration k $\big($*i.e.,* $\pi^k(\boldsymbol{c_j})\big)$. Refer to Figure 1 for an illustration. We learn $\pi^1, \ldots \pi^{k-1}$ in the prior iterations and are not trainable at iteration k. As each interpretable model g^k specializes in explaining a specific subset of samples (denoted by coverage τ), we refer to it as an *expert*. We use SelectiveNet's (Geifman & El-Yaniv, 2019) optimization method to optimize Equation (2) since selectors need a rejection mechanism to route samples through residuals. Appendix A.4 details the optimization procedure in Equation (3). We refer to the interpretable experts of all the iterations as a "Mixture of Interpretable Experts" (MoIE) cumulatively after training. Furthermore, we utilize E-LEN, *i.e.,* a Logic Explainable Network (Ciravegna et al., 2023) implemented with an Entropy Layer as first layer (Barbiero et al., 2022) as the interpretable symbolic model g to construct First Order Logic (FOL) explanations of a given prediction.

Table 1. Datasets and Blackboxes.

DATASET	BLACKBOX	# EXPERTS
CUB-200 (Wah et al., 2011)	RESNET101 (He et al., 2016)	6
CUB-200 (Wah et al., 2011)	VIT (Wang et al., 2021)	6
AWA2 (Xian et al., 2018)	RESNET101 (He et al., 2016)	4
AWA2 (Xian et al., 2018)	VIT (Wang et al., 2021)	6
HAM1000 (Tschandl et al., 2018)	INCEPTION (Szegedy et al., 2015)	6
SIIM-ISIC (Rotemberg et al., 2021)	INCEPTION (Szegedy et al., 2015)	6
EFFUSION IN MIMIC-CXR (Johnson et al.)	DENSENET121 (Huang et al., 2017)	3

2.1.3. The Residuals

The last step is to *repeat* with the residual r^k, as $r^k(\boldsymbol{x_j}, \boldsymbol{c_j}) = f^{k-1}(\boldsymbol{x_j}) - g^k(\boldsymbol{c_j})$. We train $f^k = h^k\big(\Phi(.)\big)$ to approximate the residual r^k, creating a new Blackbox f^k for the next iteration $(k+1)$. This step is necessary to specialize f^k over samples not covered by g^k. Optimizing the following loss function yields f^k for the k^{th} iteration:

$$\mathcal{L}^k_f(\boldsymbol{x_j}, \boldsymbol{c_j}) = \underbrace{\ell\big(r^k(\boldsymbol{x_j}, \boldsymbol{c_j}), f^k(\boldsymbol{x_j})\big)}_{\substack{\text{trainable component}\\ \text{for iteration } k}} \underbrace{\prod_{i=1}^{k}\big(1 - \pi^i(\boldsymbol{c_j})\big)}_{\substack{\text{non-trainable component}\\ \text{for iteration } k}} \tag{4}$$

Notice that we fix the embedding $\Phi(.)$ for all the iterations. Due to computational overhead, we only finetune the last few layers of the Blackbox (h^k) to train f^k. At the final iteration K, our method produces a MoIE and a Residual, explaining the interpretable and uninterpretable components of the initial Blackbox f^0, respectively. Appendix A.5 describes the training procedure of our model, the extraction of FOL, and the architecture of our model at inference.

Selecting number of iterations K: We follow two principles to select the number of iterations K as a stopping criterion: 1) Each expert should have enough data to be trained reliably (coverage ζ^k). If an expert covers insufficient samples, we stop the process. 2) If the final residual (r^K) underperforms a threshold, it is not reliable to distill from the Blackbox. We stop the procedure to ensure that overall accuracy is maintained.

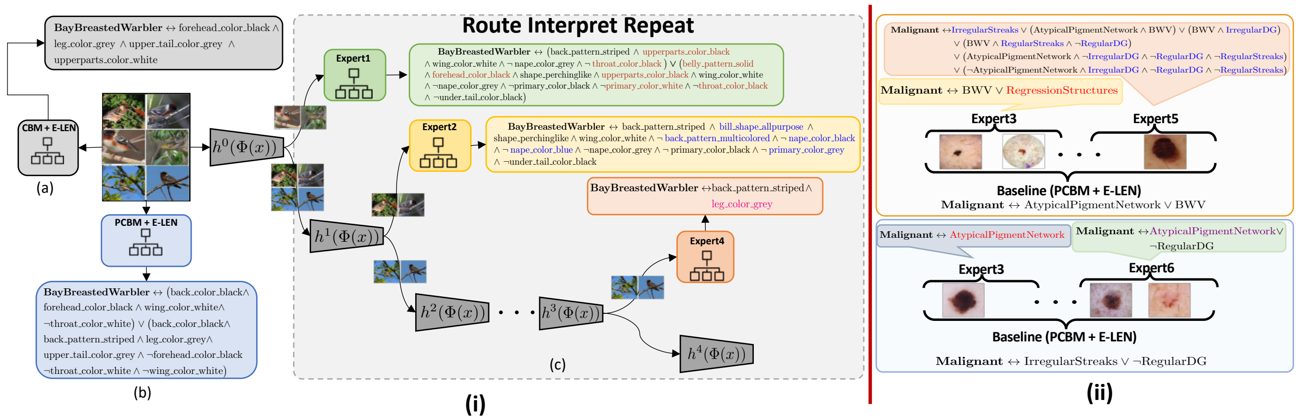

BayBreastedWarbler ↔ forehead_color_black ∧ leg_color_grey ∧ upper_tail_color_grey ∧ upperparts_color_white
CBM + E-LEN
(a)
PCBM + E-LEN
BayBreastedWarbler ↔ (back_color_black∧ forehead_color_black ∧ wing_color_white∧ ¬throat_color_white) ∨ (back_color_black∧ back_pattern_striped ∧ leg_color_grey∧ upper_tail_color_grey ∧ ¬forehead_color_black ¬throat_color_white ∧ ¬wing_color_white)
(b)
Route Interpret Repeat
$h^0(\Phi(x))$
Expert1
BayBreastedWarbler ↔ (back_pattern_striped ∧ upperparts_color_black ∧ wing_color_white ∧ ¬ nape_color_grey ∧ ¬ throat_color_black) ∨ (belly_pattern_solid ∧ forehead_color_black ∧ shape_perchinglike ∧ upperparts_color_black ∧ wing_color_white ∧ ¬nape_color_grey ∧ ¬primary_color_black ∧ ¬primary_color_white ∧ ¬throat_color_black ∧ ¬under_tail_color_black)
Expert2
BayBreastedWarbler ↔ back_pattern_striped ∧ bill_shape_allpurpose ∧ shape_perchinglike ∧ wing_color_white ∧ ¬ back_pattern_multicolored ∧ ¬ nape_color_black ∧ ¬ nape_color_blue ∧ ¬nape_color_grey ∧ ¬ primary_color_black ∧ ¬ primary_color_grey ∧ ¬under_tail_color_black
$h^1(\Phi(x))$
BayBreastedWarbler ↔back_pattern_striped∧ leg_color_grey
Expert4
$h^2(\Phi(x))$
$h^3(\Phi(x))$
$h^4(\Phi(x))$
(c)
(i)
Malignant ↔IrregularStreaks ∨ (AtypicalPigmentNetwork ∧ BWV) ∨ (BWV ∧ IrregularDG) ∨ (BWV ∧ RegularStreaks ∧ ¬RegularDG) ∨ (AtypicalPigmentNetwork ∧ ¬IrregularDG ∧ ¬RegularDG ∧ ¬RegularStreaks) ∨ (¬AtypicalPigmentNetwork ∧ IrregularDG ∧ ¬RegularDG ∧ ¬RegularStreaks)
Malignant ↔ BWV ∨ RegressionStructures
Expert3
Expert5
Baseline (PCBM + E-LEN)
Malignant ↔ AtypicalPigmentNetwork ∨ BWV
Malignant ↔ AtypicalPigmentNetwork
Malignant ↔AtypicalPigmentNetwork∨ ¬RegularDG
Expert3
Expert6
Baseline (PCBM + E-LEN)
Malignant ↔ IrregularStreaks ∨ ¬RegularDG
(ii)

Figure 2. MoIE identifies diverse concepts for specific subsets of a class, unlike the generic ones by the baselines. **(i)** We construct the FOL explanations of the samples of, "Bay breasted warbler" in the CUB-200 dataset for VIT-based **(a)** CBM + E-LEN as an *interpretable-by-design* baseline, **(b)** PCBM + E-LEN as a *posthoc* baseline, **(c)** experts in MoIE at inference. We highlight the unique concepts for experts 1,2, and 3 in *red*, *blue*, and *magenta*, respectively. **(ii)** Comparison of FOL explanations by MoIE with the PCBM + E-LEN baselines for HAM10000 (**top**) and ISIC (**down**) to classify Malignant lesion. We highlight unique concepts for experts 3, 5, and 6 in *red*, *blue*, and *violet*, respectively. For brevity, we combine the local FOLs for each expert for the samples covered by them, shown in the figure.

3. Related work

Post hoc explanations: Post hoc explanations retain the flexibility and performance of the Blackbox. The post hoc explanation has many categories, including feature attribution (Simonyan et al., 2013; Smilkov et al., 2017; Binder et al., 2016) and counterfactual approaches (Singla et al., 2019; Abid et al., 2021). For example, feature attribution methods associate a measure of importance to features (e.g., pixels) that is proportional to the feature's contribution to BlackBox's predicted output. Many methods were proposed to estimate the importance measure, including gradient-based methods (Selvaraju et al., 2017; Sundararajan et al., 2017), game-theoretic approach (Lundberg & Lee, 2017). The post hoc approaches suffer from a lack of fidelity to input (Adebayo et al., 2018) and ambiguity in explanation due to a lack of correspondence to human-understandable concepts. Recently, Posthoc Concept Bottleneck models (PCBMs) (Yuksekgonul et al., 2022) learn the concepts from a trained Blackbox embedding and use an interpretable classifier for classification. Also, they fit a residual in their hybrid variant (PCBM-h) to mimic the performance of the Blackbox. We will compare against the performance of the PCBMs method. Another major shortcoming is that, due to a lack of mechanistic explanation, post hoc explanations do not provide a recourse when an undesirable property of a Blackbox is identified. Interpretable-by-design provides a remedy to those issues (Rudin, 2019).

Concept-based interpretable models: Our approach falls into the category of concept-based interpretable models. Such methods provide a mechanistically interpretable prediction that is a function of human-understandable concepts. The concepts are usually extracted from the activation of the middle layers of the Neural Network (bottleneck). Examples include Concept Bottleneck models (CBMs) (Koh et al., 2020), antehoc concept decoder (Sarkar et al., 2022), and a high-dimensional Concept Embedding model (CEMs) (Zarlenga et al., 2022) that uses high dimensional concept embeddings to allow extra supervised learning capacity and achieves SOTA performance in the interpretable-by-design class. Most concept-based interpretable models do not model the interaction between concepts and cannot be used for reasoning. An exception is E-LEN (Barbiero et al., 2022) which uses an entropy-based approach to derive explanations in terms of FOL using the concepts. The underlying assumption of those methods is that one interpretable function can explain the entire set of data, which can limit flexibility and consequently hurt the performance of the models. Our approach relaxes that assumption by allowing multiple interpretable functions and a residual. Each function is appropriate for a portion of the data, and a small portion of the data is allowed to be uninterpretable by the model (*i.e.,* residual). We will compare our method with CBMs, CEMs, and their E-LEN-enhanced variants.

Application in fixing the shortcut learning: Shortcuts are spurious features that correlate with both input and the label on the training dataset but fail to generalize in more challenging real-world scenarios. Explainable AI (X-AI) aims to identify and fix such an undesirable property. Related work in X-AI includes LIME (Ribeiro et al., 2016), utilized to detect spurious background as a shortcut to classify an animal. Recently interpretable model (Rosenzweig et al., 2021), involving local image patches, are used as a proxy to the Blackbox to identify shortcuts. However, both methods operate in pixel space, not concept space. Also, both approaches are post hoc and do not provide a way to eliminate the shortcut learning problem. Our MoIE discovers shortcuts using the high-level concepts in the FOL explanation of the Blackbox's prediction and eliminates them via metadata normalization (MDN) (Lu et al., 2021).

4. Experiments

We perform experiments on a variety of vision and medical imaging datasets to show that 1) MoIE captures a diverse set of concepts, 2) the performance of the residuals degrades over successive iterations as they cover "harder" instances, 3) MoIE does not compromise the performance of the Blackbox, 4) MoIE achieves superior performances during test time interventions, and 5) MoIE can fix the shortcuts using the Waterbirds dataset (Sagawa et al., 2019). We repeat our method until MoIE covers at least 90% of samples or the final residual's accuracy falls below 70%. Furthermore, we only include concepts as input to g if their validation accuracy or auroc exceeds a certain threshold (in all of our experiments, we fix 0.7 or 70% as the threshold of validation auroc or accuracy). Refer to Table 1 for the datasets and Blackboxes experimented with. For convolution based Blackboxes (ResNets, Densenet121 and Inception), we flatten the feature maps from the last convolutional block to extract the concepts. For VITs, we use the image embeddings from the transformer encoder to perform the same. We use SIIM-ISIC as a real-world transfer learning setting, with the Blackbox trained on HAM10000 and evaluated on a subset of the SIIM-ISIC Melanoma Classification dataset (Yuksekgonul et al., 2022). Appendix A.6 and Appendix A.8 expand on the datasets and hyperparameters.

Baselines: We compare our methods to two concept-based baselines – 1) interpretable-by-design and 2) posthoc. They consist of two parts: a) a concept predictor $\Phi : \mathcal{X} \rightarrow \mathcal{C}$, predicting concepts from images; and b) a label predictor $g : \mathcal{C} \rightarrow \mathcal{Y}$, predicting labels from the concepts. The end-to-end CEMs and sequential CBMs serve as interpretable-by-design baselines. Similarly, PCBM and PCBM-h serve as post hoc baselines. Convolution-based Φ includes all layers till the last convolution block. VIT-based Φ consists of the transformer encoder block. The standard CBM and PCBM models do not show how the concepts are composed to make the label prediction. So, we create CBM + E-LEN, PCBM + E-LEN and PCBM-h + E-LEN by using the identical g of MOIE (shown in Appendix A.8), as a replacement for the standard classifiers of CBM and PCBM. We train the Φ and g in these new baselines to sequentially generate FOLs (Barbiero et al., 2022). Due to the unavailability of concept annotations, we extract the concepts from the Derm7pt dataset (Kawahara et al., 2018) using the pretrained embeddings of the Blackbox (Yuksekgonul et al., 2022) for HAM10000. Thus, we do not have interpretable-by-design baselines for HAM10000 and ISIC.

4.1. Results

4.1.1. Expert driven explanations by MoIE

First, we show that MoIE captures a rich set of diverse instance-specific concepts qualitatively. Next, we show quantitatively that MoIE-identified concepts are faithful to Blackbox's final prediction using the metric "completeness score" and zeroing out relevant concepts.

Heterogenity of Explanations: At each iteration of MoIE, the blackbox $\big(h^k(\Phi(.)\big)$ splits into an interpretable expert (g^k) and a residual (r^k). Figure 2i shows this mechanism for VIT-based MoIE and compares the FOLs with CBM + E-LEN and PCBM + E-LEN baselines to classify "Bay Breasted Warbler" of CUB-200. The experts of different iterations specialize in specific instances of "Bay Breasted Warbler". Thus, each expert's FOL comprises its instance-specific concepts of the same class. For example, the concept, *leg_color_grey* is unique to expert4, but *belly_pattern_solid* and *back_pattern_multicolored* are unique to experts 1 and 2, respectively, to classify the instances of "Bay Breasted Warbler" in the Figure 2(i)-c. Unlike MoIE, the baselines employ a single interpretable model g, resulting in a generic FOL with identical concepts for all the samples of "Bay Breasted Warbler" (Figure 2i(a-b)). Thus the baselines fail to capture the heterogeneity of explanations. Due to space constraint, we combine the local FOLs of different samples. For additional results of CUB-200, refer to Appendix A.10.6.

Figure 2ii shows such diverse local instance-specific explanations for HAM10000 (*top*) and ISIC (*bottom*). In Figure 2ii-(top), the baseline-FOL consists of concepts such as *AtypicalPigmentNetwork* and *BlueWhitishVeil (BWV)* to classify "Malignancy" for all the instances for HAM10000. However, expert 3 relies on *RegressionStructures* along with *BWV* to classify the same for the samples it covers while expert 5 utilizes several other concepts *e.g., IrregularStreaks*, *Irregular dots and globules (IrregularDG) etc.* Due to space constraints, Appendix A.10.7 reports similar results for the Awa2 dataset. Also, VIT-based experts compose less concepts per sample than the ResNet-based experts, shown in Appendix A.10.8.

MoIE-identified concepts attain higher completeness scores. Figure 5(a-b) shows the completeness scores (Yeh et al., 2019) for varying number of concepts. Complete-

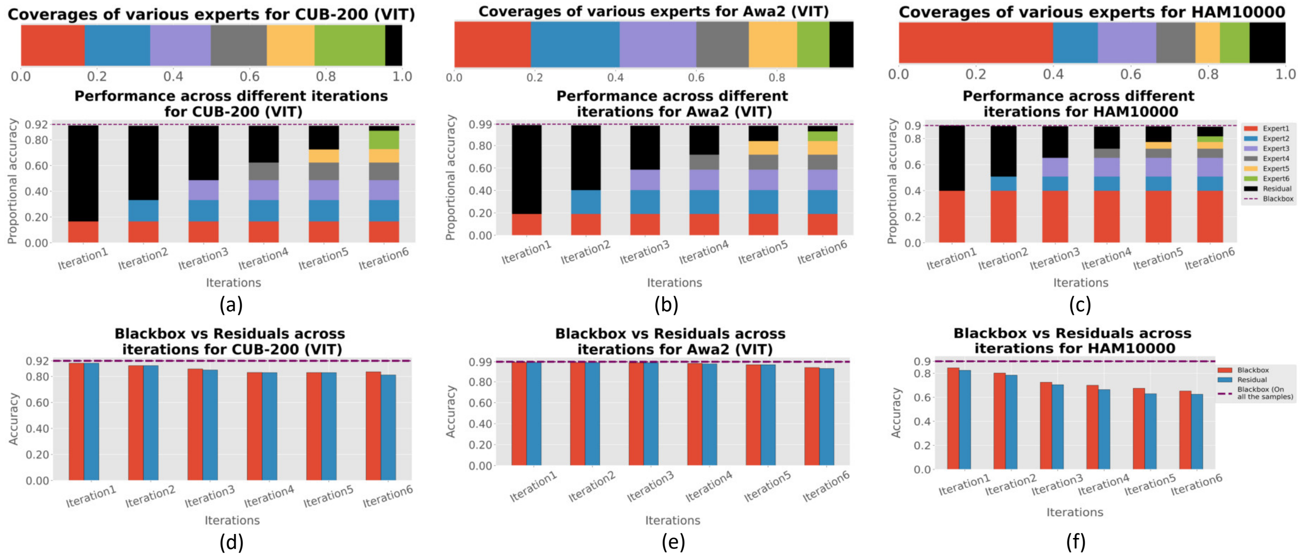


Figure 3. The performance of experts and residuals across iterations. **(a-c)** Coverage and proportional accuracy of the experts and residuals. **(d-f)** We route the samples covered by the residuals across iterations to the initial Blackbox f^0 and compare the accuracy of f^0 (red bar) with the residual (blue bar). Figures **d-f** show the progressive decline in performance of the residuals across iterations as they cover the samples in the increasing order of "hardness". We observe the similar abysmal performance of the initial blackbox f^0 for these samples.

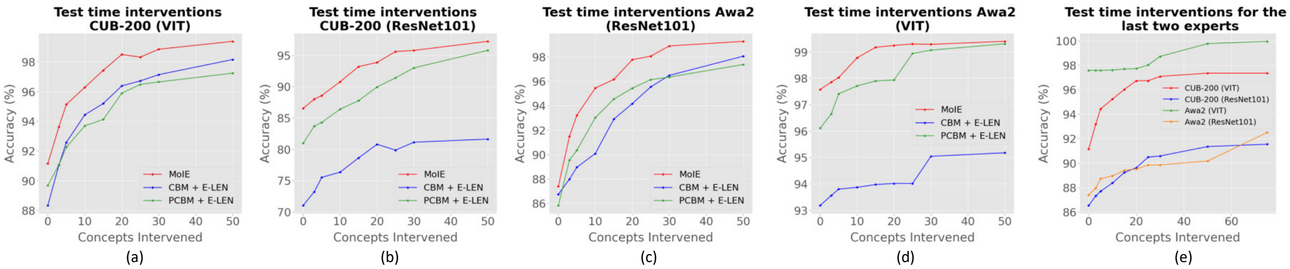


Figure 4. Across architectures test time interventions of concepts on all the samples **(a-d)**, on the "hard" samples **(e)**, covered by only the last two experts of MoIE.

ness score is a post hoc measure, signifying the identified concepts as "sufficient statistic" of the predictive capability of the Blackbox. Recall that g utilizes E-LEN (Barbiero et al., 2022), associating each concept with an attention weight after training. A concept with high attention weight implies its high predictive significance. Iteratively, we select the top relevant concepts based on their attention weights and compute the completeness scores for the top concepts for MoIE and the PCBM + E-LEN baseline in Figure 5(a-b) (Appendix A.7 for details). For example, MoIE achieves a completeness score of 0.9 compared to 0.75 of the baseline($\sim 20\% \uparrow$) for the 10 most significant concepts for the CUB-200 dataset with VIT as Blackbox.

MoIE identifies more meaningful instance-specific concepts. Figure 5(c-d) reports the drop in accuracy by zeroing out the significant concepts. Any interpretable model (g) supports concept-intervention (Koh et al., 2020). After identifying the top concepts from g using the attention weights, as in the last section, we set these concepts' values to zero, compute the model's accuracy drop, and plot in Figure 5(c-d). When zeroing out the top 10 essential concepts for VIT-based CUB-200 models, MoIE records a drop of 53% compared to 28% and 42% for the CBM + E-LEN and PCBM + E-LEN baselines, respectively, showing the faithfulness of the identified concepts to the prediction.

In both of the last experiments, MoIE outperforms the baselines as the baselines mark the same concepts as significant for all samples of each class. However, MoIE leverages various experts specializing in different subsets of samples of different classes. For results of MIMIC-CXR and Awa2,

Table 2. MoIE does not hurt the performance of the original Blackbox using a held-out test set. We provide the mean and standard errors of AUROC and accuracy for medical imaging (*e.g.,* HAM10000, ISIC, and Effusion) and vision (*e.g.,* CUB-200 and Awa2) datasets, respectively, over 5 random seeds. For MoIE, we also report the percentage of test set samples covered by all experts as "coverage". Here, MoIE + Residual represents the experts with the final residual. Following the setting (Zarlenga et al., 2022), we only report the performance of the convolutional CEM, leaving the construction of VIT-based CEM as a future work. Recall that interpretable-by-design models can not be constructed for HAM10000 and ISIC as they have no concept annotation; we learn the concepts from the Derm7pt dataset. For all the datasets, MoIE covers a significant portion of data (at least 90%) cumulatively. We boldface our results.

MODEL				**DATASET**			
	CUB-200 (RESNET101)	CUB-200 (VIT)	AWA2 (RESNET101)	AWA2 (VIT)	HAM10000	SIIM-ISIC	EFFUSION
BLACKBOX	0.88	0.92	0.89	0.99	0.96	0.85	0.91
INTERPRETABLE-BY-DESIGN							
CEM (Zarlenga et al., 2022)	0.77 ± 0.22	-	0.88 ± 0.50	-	NA	NA	0.76 ± 0.00
CBM (Sequential) (Koh et al., 2020)	0.65 ± 0.37	0.86 ± 0.24	0.88 ± 0.35	0.94 ± 0.28	NA	NA	0.79 ± 0.00
CBM + E-LEN (Koh et al., 2020; Barbiero et al., 2022)	0.71 ± 0.35	0.88 ± 0.24	0.86 ± 0.35	0.93 ± 0.25	NA	NA	0.79 ± 0.00
POSTHOC							
PCBM (Yuksekgonul et al., 2022)	0.76 ± 0.01	0.85 ± 0.20	0.82 ± 0.23	0.94 ± 0.17	0.93 ± 0.00	0.71 ± 0.01	0.81 ± 0.01
PCBM-h (Yuksekgonul et al., 2022)	0.85 ± 0.01	0.91 ± 0.18	0.87 ± 0.20	0.98 ± 0.17	0.95 ± 0.00	0.79 ± 0.05	0.87 ± 0.07
PCBM + E-LEN (Yuksekgonul et al., 2022; Barbiero et al., 2022)	0.80 ± 0.36	0.89 ± 0.26	0.85 ± 0.25	0.96 ± 0.18	0.94 ± 0.02	0.73 ± 0.01	0.81 ± 0.01
PCBM-h + E-LEN (Yuksekgonul et al., 2022; Barbiero et al., 2022)	0.85 ± 0.30	0.91 ± 0.28	0.88 ± 0.24	0.98 ± 0.20	0.95 ± 0.03	0.82 ± 0.05	0.87 ± 0.03
OURS							
MoIE (COVERAGE)	**0.86 ± 0.01 (0.9)**	**0.91 ± 0.00 (0.95)**	**0.87 ± 0.02 (0.91)**	**0.97 ± 0.00 (0.94)**	**0.95 ± 0.00 (0.9)**	**0.84 ± 0.00 (0.94)**	**0.87 ± 0.00 (0.98)**
MoIE + RESIDUAL	**0.84 ± 0.01**	**0.90 ± 0.01**	**0.86 ± 0.020**	**0.94 ± 0.004**	**0.92 ± 0.00**	**0.82 ± 0.01**	**0.86 ± 0.00**

refer to Appendix A.10.2 and Appendix A.10.4 respectively.

4.1.2. Identification of harder samples by successive residuals

Figure 3 (a-c) display the proportional accuracy of the experts and the residuals of our method per iteration. The proportional accuracy of each model (experts and/or residuals) is defined as the accuracy of that model times its coverage. Recall that the model's coverage is the empirical mean of the samples selected by the selector. Figure 3a show that the experts and residual cumulatively achieve an accuracy ~ 0.92 for the CUB-200 dataset in iteration 1, with more contribution from the residual (black bar) than the expert1 (blue bar). Later iterations cumulatively increase and worsen the performance of the experts and corresponding residuals, respectively. The final iteration carves out the entire interpretable portion from the Blackbox f^0 via all the experts, resulting in their more significant contribution to the cumulative performance. The residual of the last iteration covers the "hardest" samples, achieving low accuracy. Tracing these samples back to the original Blackbox f^0, it also classifies these samples poorly (Figure 3(d-f)). As shown in the coverage plot, this experiment reinforces Figure 1, where the flow through the experts gradually becomes thicker compared to the narrower flow of the residual with every iteration. Refer to Figure 12 in the Appendix A.10.3 for the results of the ResNet-based MoIEs.

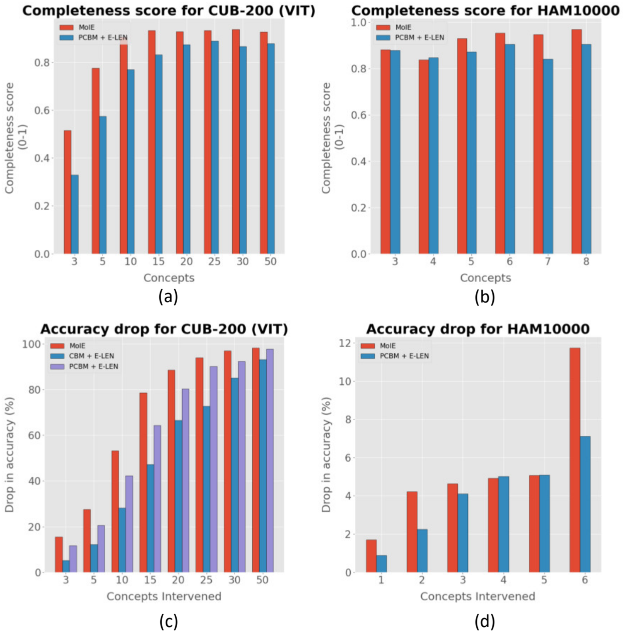
4.1.3. Quantitative analysis of MoIE with the Blackbox and baseline

Comparing with the interpretable-by-design baselines: Table 2 shows that MoIE achieves comparable performance to the Blackbox. Recall that "MoIE" refers to the mixture of all interpretable experts (g) only excluding any residuals. MoIE outperforms the interpretable-by-design baselines for all the datasets except Awa2. Since Awa2 is designed for zero-shot learning, its rich concept annotation makes it appropriate for interpretable-by-design models. In general, VIT-derived MoIEs perform better than their ResNet-based variants.

Comparing with the PCBMs: Table 2 shows that interpretable MoIE outperforms the interpretable posthoc baselines – PCBM and PCBM + E-LEN for all the datasets, especially by a significant margin for CUB-200 and ISIC. We also report "MoIE + Residual" as the mixture of interpretable experts plus the final residual to compare with the residualized PCBM, *i.e.,* PCBM-h. Table 2 shows that PCBM-h performs slightly better than MoIE + Residual. Note that PCBM-h learns the residual by fitting the complete dataset to fix the interpretable PCBM's mistakes to replicate the performance of the Blackbox, resulting in better performance for PCBM-h than PCBM. However, we assume the Blackbox to be a combination of interpretable and uninterpretable components. So, we train the experts and the final residual to cover the interpretable and uninterpretable portions of the Blackbox respectively. In each iteration, our method learns the residuals to focus on the samples, which are not covered by the respective interpretable experts. Therefore, residuals are not designed to fix the mistakes made by the experts. In doing so, the final residual in MoIE + Residual covers the "hardest" examples, lowering its overall performance compared to MoIE.

4.1.4. Test time interventions

Figure 4(a-d) shows effect of test time interventions. Any concept-based models (Koh et al., 2020; Zarlenga et al.,



(a)

(b)

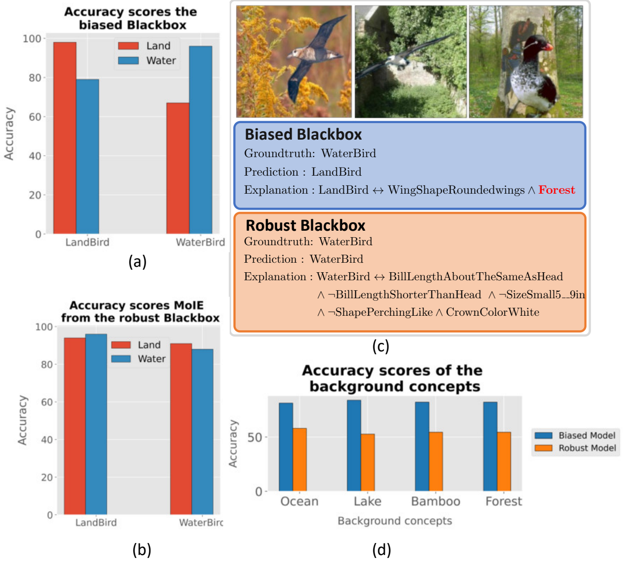
(c)

(d)

Figure 5. Quantitative validation of the extracted concepts. **(a-b)** Completeness scores of the models for a varying number of top concepts. **(c-d)** Drop in accuracy compared to the original model after zeroing out the top significant concepts iteratively. The highest drop for MoIE indicates that MoIE selects more instance-specific concepts than generic ones by the baselines.

2022) allow test time interventions for datasets with concept annotation (*e.g.,* CUB-200, Awa2). We identify the significant concepts via their attention scores in g, as during the computation of completeness scores, and set their values with the ground truths, considering the ground truth concepts as an oracle. As MoIE identifies a more diverse set of concepts by focusing on different subsets of classes, MoIE outperforms the baselines in terms of accuracy for such test time interventions. Instead of manually deciding the samples to intervene, it is generally preferred to intervene on the "harder" samples, making the process efficient. As per Section 4.1.2, experts of different iterations cover samples with increasing order of "hardness". To intervene efficiently, we perform identical test-time interventions with varying numbers of concepts for the "harder" samples covered by the final two experts and plot the accuracy in Figure 4(e). For the VIT-derived MoIE of CUB-200, intervening only on 20 concepts enhances the accuracy of MoIE from 91% to 96% ($\sim$ 6.1% $\uparrow$). We cannot perform the same for the baselines as they cannot directly estimate "harder" samples. Also, Figure 4 shows a relatively higher gain for ResNet-based models in general. Appendix A.10.5 demonstrates an example of test time intervention of concepts for relatively "harder" samples, identified by the last two experts of MoIE.

4.1.5. Application in the removal of shortcuts



(a)

(c)

(b)

(d)

Figure 6. MoIE fixes shortcuts. **(a)** Performance of the biased Blackbox. **(b)** Performance of final MoIE extracted from the robust Blackbox after removing the shortcuts using MDN. **(c)** Examples of samples (**top-row**) and their explanations by the biased (**middle-row**) and robust Blackboxes (**bottom-row**). **(d)** Comparison of accuracies of the spurious concepts extracted from the biased vs. the robust Blackbox.

First, we create the Waterbirds dataset as in (Sagawa et al., 2019)by using forest and bamboo as the spurious land concepts of the Places dataset for landbirds of the CUB-200 dataset. We do the same by using oceans and lakes as the spurious water concepts for waterbirds. We utilize ResNet50 as the Blackbox f^0 to identify each bird as a Waterbird or a Landbird. The Blackbox quickly latches on the spurious backgrounds to classify the birds. As a result, the black box's accuracy differs for land-based versus aquatic subsets of the bird species, as shown in Figure 6a. The Waterbird on the water is more accurate than on land (96% vs. 67% in the red bar in the Figure 6a). The FOL from the biased Blackbox-derived MoIE captures the spurious concept *forest* for a waterbird, misclassified as a landbird. Assuming the background concepts as metadata, we remove the background bias from the representation of the Blackbox using Metadata normalization (MDN) layers (Lu et al., 2021) between two successive layers of the convolutional backbone to fine-tune the biased Blackbox to make it more robust. Next, we train t, using the embedding Φ of the robust Blackbox, and compare the accuracy of the spurious concepts with the biased blackbox in Figure 6d. The validation accuracy of all the spurious concepts retrieved from the robust Blackbox falls well short of the predefined threshold 70% compared to the biased Blackbox. Finally, we re-train the MoIE distilling from the new robust Blackbox. Figure 6b illustrates similar accuracies of MoIE for Waterbirds on water vs. Waterbirds on land (89% - 91%). The FOL from the robust Blackbox does not include any background concepts (6c, bottom row). Refer to 8 in Appendix A.9 for the flow diagram of this experiment.

5. Discussion & Conclusions

This paper proposes a novel method to iteratively extract a mixture of interpretable models from a flexible Blackbox. The comprehensive experiments on various datasets demonstrate that our method 1) captures more meaningful instance-specific concepts with high completeness score than baselines without losing the performance of the Blackbox, 2) does not require explicit concept annotation, 3) identifies the "harder" samples using the residuals, 4) achieves significant performance gain than the baselines during test time interventions, 5) eliminate shortcuts effectively. In the future, we aim to apply our method to other modalities, such as text or video. Also, as in the prior work, MoIE-captured concepts may not reflect a causal effect. The assessment of causal concept effects necessitates estimating inter-concept interactions, which will be the subject of future research.

6. Acknowledgement

We would like to thank Mert Yuksekgonul of Stanford University for providing the code to construct the concept bank of Derm7pt for conducting the skin experiments. This work was partially supported by NIH Award Number 1R01HL141813-01 and the Pennsylvania Department of Health. We are grateful for the computational resources provided by Pittsburgh Super Computing grant number TG-ASC170024.

References

Abid, A., Yuksekgonul, M., and Zou, J. Meaningfully explaining model mistakes using conceptual counterfactuals. *arXiv preprint arXiv:2106.12723*, 2021.

Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., and Kim, B. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018.

Alharbi, R., Vu, M. N., and Thai, M. T. Learning interpretation with explainable knowledge distillation. In *2021 IEEE International Conference on Big Data (Big Data)*, pp. 705–714. IEEE, 2021.

Barbiero, P., Ciravegna, G., Giannini, F., Lió, P., Gori, M., and Melacci, S. Entropy-based logic explanations of neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 6046–6054, 2022.

Belle, V. Symbolic logic meets machine learning: A brief survey in infinite domains. In *International Conference on Scalable Uncertainty Management*, pp. 3–16. Springer, 2020.

Besold, T. R., Garcez, A. d., Bader, S., Bowman, H., Domingos, P., Hitzler, P., Kühnberger, K.-U., Lamb, L. C., Lowd, D., Lima, P. M. V., et al. Neural-symbolic learning and reasoning: A survey and interpretation. *arXiv preprint arXiv:1711.03902*, 2017.

Binder, A., Montavon, G., Lapuschkin, S., Müller, K.-R., and Samek, W. Layer-wise relevance propagation for neural networks with local renormalization layers. In *International Conference on Artificial Neural Networks*, pp. 63–71. Springer, 2016.

Breiman, L., Friedman, J., Stone, C., and Olshen, R. Classification and regression trees (crc, boca raton, fl). 1984.

Cheng, X., Rao, Z., Chen, Y., and Zhang, Q. Explaining knowledge distillation by quantifying the knowledge. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12925–12935, 2020.

Ciravegna, G., Barbiero, P., Giannini, F., Gori, M., Lió, P., Maggini, M., and Melacci, S. Logic explained networks. *arXiv preprint arXiv:2108.05149*, 2021.

Ciravegna, G., Barbiero, P., Giannini, F., Gori, M., Lió, P., Maggini, M., and Melacci, S. Logic explained networks. *Artificial Intelligence*, 314:103822, 2023.

Daneshjou, R., Vodrahalli, K., Liang, W., Novoa, R. A., Jenkins, M., Rotemberg, V., Ko, J., Swetter, S. M., Bailey, E. E., Gevaert, O., et al. Disparities in dermatology ai: Assessments using diverse clinical images. *arXiv preprint arXiv:2111.08006*, 2021.

Garcez, A. d., Besold, T. R., De Raedt, L., Földiak, P., Hitzler, P., Icard, T., Kühnberger, K.-U., Lamb, L. C., Miikkulainen, R., and Silver, D. L. Neural-symbolic learning and reasoning: contributions and challenges. In *2015 AAAI Spring Symposium Series*, 2015.

Geifman, Y. and El-Yaniv, R. Selectivenet: A deep neural network with an integrated reject option. In *International conference on machine learning*, pp. 2151–2159. PMLR, 2019.

Hastie, T. and Tibshirani, R. Generalized additive models: some applications. *Journal of the American Statistical Association*, 82(398):371–386, 1987.

Havasi, M., Parbhoo, S., and Doshi-Velez, F. Addressing leakage in concept bottleneck models. In *Advances in Neural Information Processing Systems*, 2022.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hinton, G., Vinyals, O., Dean, J., et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Jain, S., Agrawal, A., Saporta, A., Truong, S. Q., Duong, D. N., Bui, T., Chambon, P., Zhang, Y., Lungren, M. P., Ng, A. Y., et al. Radgraph: Extracting clinical entities and relations from radiology reports. *arXiv preprint arXiv:2106.14463*, 2021.

Johnson, A., Lungren, M., Peng, Y., Lu, Z., Mark, R., Berkowitz, S., and Horng, S. Mimic-cxr-jpg-chest radiographs with structured labels.

Kawahara, J., Daneshvar, S., Argenziano, G., and Hamarneh, G. Seven-point checklist and skin lesion classification using multitask multimodal neural nets. *IEEE journal of biomedical and health informatics*, 23(2):538–546, 2018.

Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., and Sayres, R. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav).(2017). *arXiv preprint arXiv:1711.11279*, 2017.

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

Letham, B., Rudin, C., McCormick, T. H., and Madigan, D. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9(3):1350–1371, 2015.

Lu, M., Zhao, Q., Zhang, J., Pohl, K. M., Fei-Fei, L., Niebles, J. C., and Adeli, E. Metadata normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10917–10927, 2021.

Lucieri, A., Bajwa, M. N., Braun, S. A., Malik, M. I., Dengel, A., and Ahmed, S. On interpretability of deep learning based skin lesion classifiers using concept activation vectors. In *2020 international joint conference on neural networks (IJCNN)*, pp. 1–10. IEEE, 2020.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*, pp. 4768–4777, 2017.

Mendelson, E. *Introduction to mathematical logic*. Chapman and Hall/CRC, 2009.

Montavon, G., Samek, W., and Müller, K.-R. Methods for interpreting and understanding deep neural networks. *Digital signal processing*, 73:1–15, 2018.

Ribeiro, M. T., Singh, S., and Guestrin, C. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

Rosenzweig, J., Sicking, J., Houben, S., Mock, M., and Akila, M. Patch shortcuts: Interpretable proxy models efficiently find black-box vulnerabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 56–65, 2021.

Rotemberg, V., Kurtansky, N., Betz-Stablein, B., Caffery, L., Chousakos, E., Codella, N., Combalia, M., Dusza, S., Guitera, P., Gutman, D., et al. A patient-centric dataset of images and metadata for identifying melanomas using clinical context. *Scientific data*, 8(1):1–8, 2021.

Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Samek, W., Binder, A., Montavon, G., Lapuschkin, S., and Müller, K.-R. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11):2660–2673, 2016.

Sarkar, A., Vijaykeerthy, D., Sarkar, A., and Balasubramanian, V. N. A framework for learning ante-hoc explainable models via concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10286–10295, 2022.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Shrikumar, A., Greenside, P., Shcherbina, A., and Kundaje, A. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*, 2016.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Singla, S., Pollack, B., Chen, J., and Batmanghelich, K. Explanation by progressive exaggeration. *arXiv preprint arXiv:1911.00483*, 2019.

Smilkov, D., Thorat, N., Kim, B., Viégas, F., and Wattenberg, M. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Tschandl, P., Rosendahl, C., and Kittler, H. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1):1–9, 2018.

Wadden, D., Wennberg, U., Luan, Y., and Hajishirzi, H. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5784–5789, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1585. URL https://aclanthology.org/D19-1585.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The caltech-ucsd birds-200-2011 dataset. 2011.

Wan, C., Belo, R., and Zejnilovic, L. Explainability's gain is optimality's loss? how explanations bias decision-making. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 778–787, 2022.

Wang, J., Yu, X., and Gao, Y. Feature fusion vision transformer for fine-grained visual categorization. *arXiv preprint arXiv:2107.02341*, 2021.

Xian, Y., Lampert, C. H., Schiele, B., and Akata, Z. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2251–2265, 2018.

Yeh, C.-K., Kim, B., Arik, S., Li, C.-L., Ravikumar, P., and Pfister, T. On concept-based explanations in deep neural networks. 2019.

Yu, K., Ghosh, S., Liu, Z., Deible, C., and Batmanghelich, K. Anatomy-guided weakly-supervised abnormality localization in chest x-rays. *arXiv preprint arXiv:2206.12704*, 2022.

Yuksekgonul, M., Wang, M., and Zou, J. Post-hoc concept bottleneck models. *arXiv preprint arXiv:2205.15480*, 2022.

Zarlenga, M. E., Barbiero, P., Ciravegna, G., Marra, G., Giannini, F., Diligenti, M., Shams, Z., Precioso, F., Melacci, S., Weller, A., et al. Concept embedding models. *arXiv preprint arXiv:2209.09056*, 2022.

A. Appendix

A.1. Project page

Refer to the url `https://shantanu48114860.github.io/projects/ICML-2023-MoIE/` for the details of this project.

A.2. Background of First-order logic (FOL) and Neuro-symbolic-AI

FOL is a logical function that accepts predicates (concept presence/absent) as input and returns a True/False output being a logical expression of the predicates. The logical expression, which is a set of AND, OR, Negative, and parenthesis, can be written in the so-called Disjunctive Normal Form (DNF) (Mendelson, 2009). DNF is a FOL logical formula composed of a disjunction (OR) of conjunctions (AND), known as the "sum of products".

Neuro-symbolic AI is an area of study that encompasses deep neural networks with symbolic approaches to computing and AI to complement the strengths and weaknesses of each, resulting in a robust AI capable of reasoning and cognitive modeling (Belle, 2020). 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).

A.3. Learning the concepts

As discussed in Section 2, $f^0 : \mathcal{X} \rightarrow \mathcal{Y}$ is a pre-trained Blackbox. Also, $f^0(.) = h^0 \circ \Phi(.)$. Here, $\Phi : \mathcal{X} \rightarrow R^l$ is the image embeddings, transforming the input images to an intermediate representation and $h^0 : R^l \rightarrow \mathcal{Y}$ is the classifier, classifying the output $\mathcal{Y}$ using the embeddings, Φ. Our approach is applicable for both datasets with and without human-interpretable concept annotations. For datasets with the concept annotation $\mathcal{C} \in \mathbb{R}^{N_c}$ (N_c being the number of concepts per image $\mathcal{X}$), we learn $t : R^l \rightarrow \mathcal{C}$ to classify the concepts using the embeddings. Per this definition, t outputs a scalar value c representing a single concept for each input image. We adopt the concept learning strategy in PosthocCBM (PCBM) (Yuksekgonul et al., 2022) for datasets without concept annotation. Specifically, we leverage a set of image embeddings with the concept being present and absent. Next, we learn a linear SVM (t) to construct the concept activation matrix (Kim et al., 2017) as $\boldsymbol{Q} \in \mathbb{R}^{N_c \times l}$. Finally we estimate the concept value as $c = \frac{<\Phi(x), q^i>}{||q_i||_2^2} \in \mathbb{R}$ utilizing each row $\boldsymbol{q^i}$ of $\boldsymbol{Q}$. Thus, the complete tuple of j^{th} sample is $\{x_j, y_j, c_j\}$, denoting the image, label, and learned concept vector, respectively.

A.4. Optimization

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

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

where τ is the target coverage and λ_s is a hyperparameter (Lagrange multiplier). We define $\mathcal{R}(.)$ and $\mathcal{L}_{g,\pi}(.)$ in Equation (1) and Equation (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.5. Algorithm

Algorithm 1 explains the overall training procedure of our method. Figure 7 displays the architecture of our model in iteration k.

A.6. Dataset

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 (Barbiero et al., 2022) 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^0.

Animals with attributes2 (Awa2) AwA2 dataset (Xian et al., 2018) consists of 37322 images of total 50 animals classes with 85 numeric attribute. 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^0.

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^0. We follow the strategy in (Lucieri et al., 2020) to extract the 8 concepts from the Derm7pt ((Kawahara et al., 2018)) dataset.

SIIM-ISIC To test a real-world transfer learning use case, we evaluate the model trained on HAM10000 on a subset of the SIIM-ISIC(Rotemberg et al., 2021)) Melanoma Classification dataset. We use the same concepts described in the HAM10000 dataset.

MIMIC-CXR We use 220,763 frontal images from the MIMIC-CXR dataset (Johnson et al.) aiming to classify effusion. We obtain the anatomical and observation concepts from the RadGraph annotations in RadGraph's inference dataset ((Jain et al., 2021)), automatically 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^0.

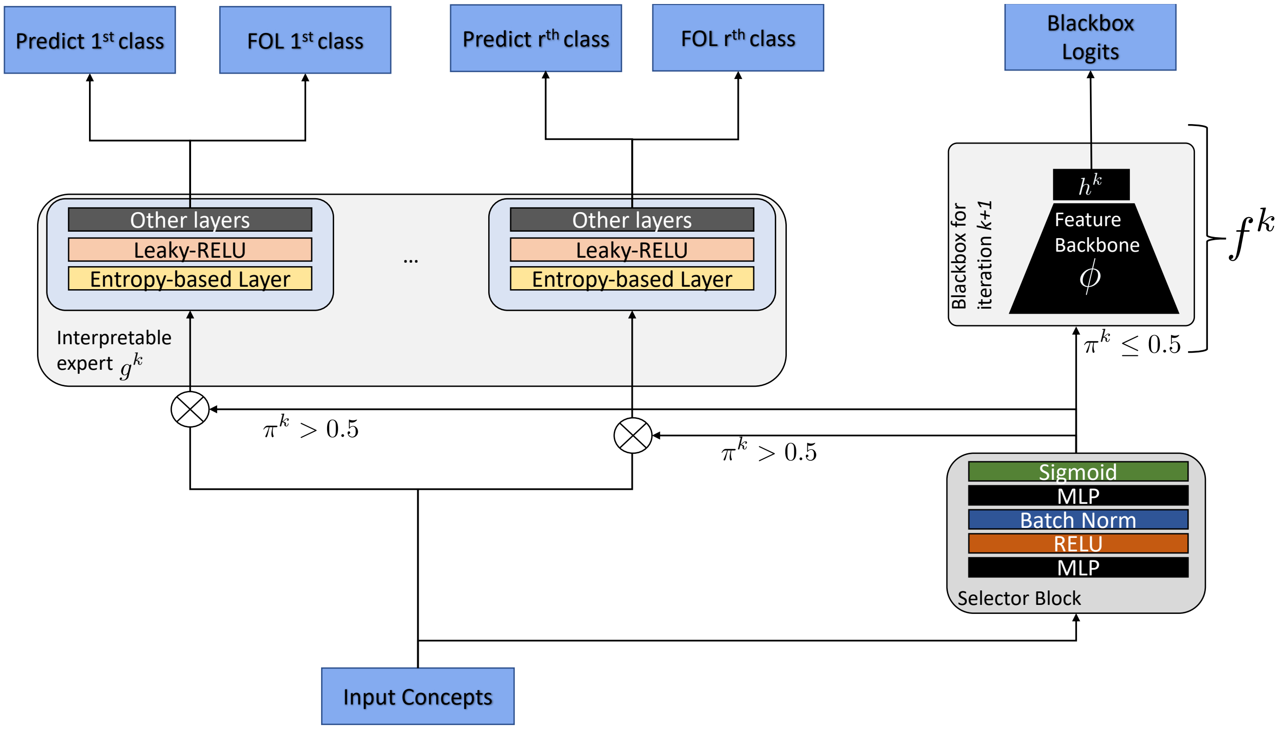


Figure 7. Architecture of MoIE. In an iteration k during inference, the selector routes the samples to go through the interpretable expert g^k if the probability $\pi^k \geq 0.5$. If $\pi^k < 0.5$, the selector routes the samples, through f^k, the Blackbox for iteration $k+1$. Note $f^k = h^k(\Phi(.)$ is an approximation of the residual $r^k = f^{k-1} - g^k$.

A.7. Estimation of completeness score

Let $f^0(x) = h^0(\Phi(\boldsymbol{x})$ is the initial Blackbox as per Section 2. The Concept completeness paper (Yeh et al., 2019) assumes $\Phi(\boldsymbol{x}) \in \mathbb{R}^l$ (*s.t.,* $l = T.d$) to be a concatanation of $[\phi(\boldsymbol{x}_1), \phi(\boldsymbol{x}_2), \dots, \phi(\boldsymbol{x}_T)]$ *s.t.,* $\phi(\boldsymbol{x}) \in \mathbb{R}^d$. Recall we utilize t to learn the concepts $\mathcal{C}$ with N_c being the total number of concepts per image. So the parameters of t, represented by $\omega_1, \omega_2, \dots \omega_{N_c}$ *s.t.,* $\omega_i \in \mathbb{R}^d$ represent linear direction in the embedding space $\phi(.) \in \mathbb{R}^d$. Next, we compute the concept product $v_c(\boldsymbol{x}_t)(< \phi(\boldsymbol{x}_t), \omega_j >)_{j=1}^{N_c}$, denoting the similarity between the image embedding and linear direction of j^{th} concept. Finally, we normalize $v_c(.)$ to obtain the concept score as $v_v(\boldsymbol{x}) = \big(\frac{v_c(\boldsymbol{x_t})}{||v_c(\boldsymbol{x}_t)||_2}\big)_{t=1}^T \in \mathbb{R}^{T.N_c}$.

Next for a Blackbox $f^0(x) = h^0(\Phi(\boldsymbol{x})$, set of concepts $c_1, c_2, \dots c_{N_c}$ and their linear direction $\omega_1, \omega_2, \dots \omega_{N_c}$ in the embedding space and, we compute the completeness score as:

$$\eta_{f^0} = \frac{\sup_\Gamma \mathbb{P}_{\boldsymbol{x},y\sim V}[y = \arg\max_{y'} h^0_{y'}(\Gamma(v_c(\boldsymbol{x})))] - a_r}{\mathbb{P}_{\boldsymbol{x},y\sim V}[y = \arg\max_{y'} f^0_{y'}(\boldsymbol{x})] - a_r}, \tag{10}$$

where V is the validation set and $\Gamma : \mathbb{R}^{T.m} \to \mathbb{R}^l$, projection from the concept score to the embedding spaceΦ. For CUB-200 and Awa2 we estimate $\mathbb{P}_{\boldsymbol{x},y\sim V}[y = \arg\max_{y'} h^0_{y'}(\Gamma(v_c(\boldsymbol{x})))]$ as the best accuracy using the given concepts and a_r is the random accuracy. For HAM10000, we estimate the same as the best AUROC. Completeness score indicates the consistency between the prediction based just on concepts and the given Blackboxf^0. If the identified concepts are sufficiently rich, label prediction will be similar to the Blackbox, resulting in higher completeness scores for the concept set. In all our experiments, Γ is a two-layer feedforward neural network with 1000 neurons.

To plot the completeness score in Figure 5a-c, we select the topN concepts iteratively representing the $N < N_c$ concepts most significant to the prediction of the interpretable model g. Recall we follow Entropy based linear neural network (Barbiero

Algorithm 1 *Route, interpret* and *repeat* algorithm to generate FOL explanations locally.

```
Input: Complete tuple: $\{x_j, y_j, c_j\}_{j=1}^{n}$; initial blackbox $f^0 = h^0(\Phi(.))$; K as the total iterations; Coverages $\tau_1, \dots, \tau_K$.
Output: Sparse mixture of experts and their selectors $\{g^k, \pi^k\}_{k=1}^{K}$ and the final residual $f^K = h^K(\Phi(.))$
Fix $\Phi$.
for $k = 1 \dots K$ do
    Fix $\pi^1 \dots \pi^{k-1}$.
    Minimize $\mathcal{L}^k$ using equation 9 to learn $\pi^k$ and $g^k$.
    Calculate $r^k = f^{k-1}(.) - g^k(.)$
    Minimize equation 4 to learn $f^k(.)$, the new blackbox for the next iteration $k + 1$.
end for
for $k = 1 \dots K$ do
    for sample $j$ in test-set do
        repeat
            Initialize sub_select_concept = $True$
            Initialize the percentile_threshold = 99.
            Retrieve the predicted class label of sample $j$ from the expert $k$ as: $\hat{y}_j = g^k(c_j)$
            Create a mask vector $m_j$. $m_j[i] = 1$ if $\tilde{\alpha}[\hat{y}_j][i] \geq$ percentile($\tilde{\alpha}[\hat{y}_j]$, percentile_threshold) and 0 otherwise. Specifically, the $i^{th}$ entry in $m_j$ is one if the $i^{th}$ value of the attention score $\tilde{\alpha}[\hat{y}_j]$ is greater than (percentile_attention)$^{th}$ percentile.
            Subselect the concept vector as $\tilde{c}_j$ as: $\tilde{c}_j = c_j \odot m_j$
            if $g^k(\tilde{c}_j) \neq \hat{y}_j$ then
                percentile_threshold = percentile_threshold - 1
                sub_select_concept = $false$
            end if
        until sub_select_concept is $True$
        Using the subselected concept vector $\tilde{c_j}$, construct the FOL expression of the $j^{th}$ sample as suggested by (Barbiero et al., 2022).
    end for
end for
```

et al., 2022) as g. So each concept has an associated attention score, α in g (Barbiero et al., 2022), denoting the importance of the concept for the prediction. We select the topN concepts based on the N concepts with highest attention weights. We get the linear direction of these topN concepts from the parameters of the learned t and project it to the embedding space Φ using Γ. If Γ reconstructs the discriminative features from the concepts successfully, the concepts achieves high completeness scores, showing faithfulness with the Blackbox. Recall Figure 5a-c demonstrate that MoIE outperforms the baselines in terms of the completeness scores. This suggests that MoIE identifies rich instance-specific concepts than the baselines, being consistent with the Blackbox.

A.8. Architectural details of symbolic experts and hyperparameters

Table 3 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 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). Table 4, Table 5, Table 6, Table 7 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.

Table 3. 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	512
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

Table 7. Hyperparameter setting of interpretable experts (g) for the dataset MIMIC-CXR

Settings based on dataset	**Expert1**	**Expert2**	**Expert3**
Effusion-MIMIC-CXR (DenseNet-121)			
+ Batch size	1028	1028	1028
+ Coverage (τ)	0.6	0.2	0.15
+ 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
+T_{lens}	7.6	7.6	7.6

A.9. Flow diagram to eliminate shotcut

Figure 8 shows the flow digram to eliminate shortcut.

A.10. More Results

A.10.1. Comparison with other interpretable by design baselines

Table 8. Comparing the performance of MoIE with additional interpretable by design baselines.

MODEL	DATASET		
	CUB-200 (RESNET101)	AWA2 (RESNET101)	EFFUSION
BLACKBOX	0.88	0.89	0.91
ANTEHOC W SUP (Sarkar et al., 2022)	0.71	0.85	0.75
ANTEHOC W/O SUP (Sarkar et al., 2022)	0.64	0.81	0.70
HARD W AR (Havasi et al., 2022)	0.81	0.84	0.73
HARD W/O AR (Havasi et al., 2022)	0.78	0.81	0.71
OURS			
MoIE (COVERAGE)	**0.86**	**0.87**	**0.87**
MoIE + RESIDUAL	**0.84**	**0.86**	**0.86**

Table 8 compares our method with several other interpretable by design baselines.

Table 4. Hyperparameter setting of interpretable experts (g) trained on ResNet-101 (top) and VIT (bottom) blackboxes for the CUB-200 dataset

Settings based on dataset	**Expert1**	**Expert2**	**Expert3**	**Expert4**	**Expert5**	**Expert6**
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
+ 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
+T_{lens}	6.0	6.0	6.0	6.0	6.0	6.0

A.10.2. Results of Effusion of MIMIC-CXR

Figure 9 demonstrates the diversity of instance-specific local FOL explanations of different concepts of MoIE and the final residual. Figure 10(a) shows the completeness scores for different concepts. Figure 10(b) shows the drop in AUROC while zeroing out different concepts. Figure 11(a) shows test time interventions of different concepts on all samples. Figure 11(b) shows test time interventions of different concepts on only the "hard" samples covered by the last two experts.

A.10.3. Performance of experts and residual for ResNet-derived experts of Awa2 and CUB-200 datasets

Figure 12 shows the coverage (top row), performances (bottom row) of each expert and residual across iterations of - (a) ResNet101-derived Awa2 and (b) ResNet101-derived CUB-200 respectively.

A.10.4. Concept validation of Awa2

Figure 13 shows the completeness scores and the drop in accuracy by zeroing out the concepts for Awa2.

A.10.5. Example of expert-specific test time intervention

Figure 14 demonstrates an example of test time intervention of concepts for "harder" samples identified by the last two experts of VIT-driven MoIE.

A.10.6. Diversity of explanations for CUB

Figure 15 shows the construction of instance-specific local FOL explanations of a category, "Olive sided Flycatcher" in the CUB-200 dataset for the VIT-based baselines and MoIE. In this example, the final expert6 covers the relatively "harder" sample. Figure 16, Figure 17, Figure 18, Figure 19 shows more such FOL explanations. All these examples demonstrate MoIE's high capability to identify more meaningful instance-specific concepts in FOL explanations. In contrast, the baselines identify the generic concepts for all samples in a class.

Table 5. Hyperparameter setting of interpretable experts (g) trained on ResNet-101 (top) and VIT (bottom) blackboxes for the Awa2 dataset

Settings based on dataset	**Expert1**	**Expert2**	**Expert3**	**Expert4**	**Expert5**	**Expert6**
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	-	-
+ 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
+ T_{lens}	6.0	6.0	6.0	6.0	6.0	6.0

A.10.7. Diversity of explanations for Awa2

Figure 20 and 21 demonstrate the flexibility of instance-specific local FOL explanations by VIT-derived MoIE compared to the different baselines for the Awa2 dataset qualitatively.

A.10.8. VIT-based experts compose of less concepts than the ResNet-based counterparts

Figure 22 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 22a). The 105 concepts by 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 22b). 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 24 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, Figure 23 (b) and Figure 25 (b) reveal that experts 2 and 6 require 25 and 58 concepts on average to explain "Artic_Tern" correctly respectively for VIT-derived MoIE.

Figure 29, Figure 30, Figure 31 display the average number of concepts required to predict an animal species correctly in the Awa2 dataset for VIT as backbones. Similarly Figure 32 and Figure 33 display the average number of concepts required to predict an animal species correctly in the Awa2 dataset for ResNet101 as backbones. We can see that for ResNet101, on average, 80 concepts are required to explain a sample correctly for the class "Weasel" (Expert1 in Figure 32 a). However, for VIT, only three concepts are needed to explain a sample correctly for "Weasel" (Expert 6 in Figure 31 f). Also, from both of these figures, we can see that different experts require different number concepts to explain the same class. For example, Figure 31 (e) and (f) reveal that experts 5 and 6 require 4 and 30 concepts on average to explain "Wolf" correctly.

Table 6. Hyperparameter setting of interpretable experts (g) for the dataset HAM10000

Settings based on dataset	**Expert1**	**Expert2**	**Expert3**	**Expert4**	**Expert5**	**Expert6**
HAM10000 (Inception-V3)						
+ Batch size	32	32	32	32	32	32
+ Coverage (τ)	0.4	0.2	0.2	0.2	0.1	0.1
+ 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	64	64	64	64	64	64
+ T_{lens}	0.7	0.7	0.7	0.7	0.7	0.7

A.11. Computational performance

Figure 34 shows the computational performance compared to the Blackbox. Though in MoIE, we sequentially learn the experts and the residuals, they take less computational resources than the Blackbox. The experts are shallow neural networks. Also, we only update the classification layer (h) for the residuals, so it takes such less time. The Flops in the Y axis are computed as Flop of (forward propagation + backward propagation) × (minibatch size) × (no of training epochs). We use the Pytorch profiler package to monitor the flops.

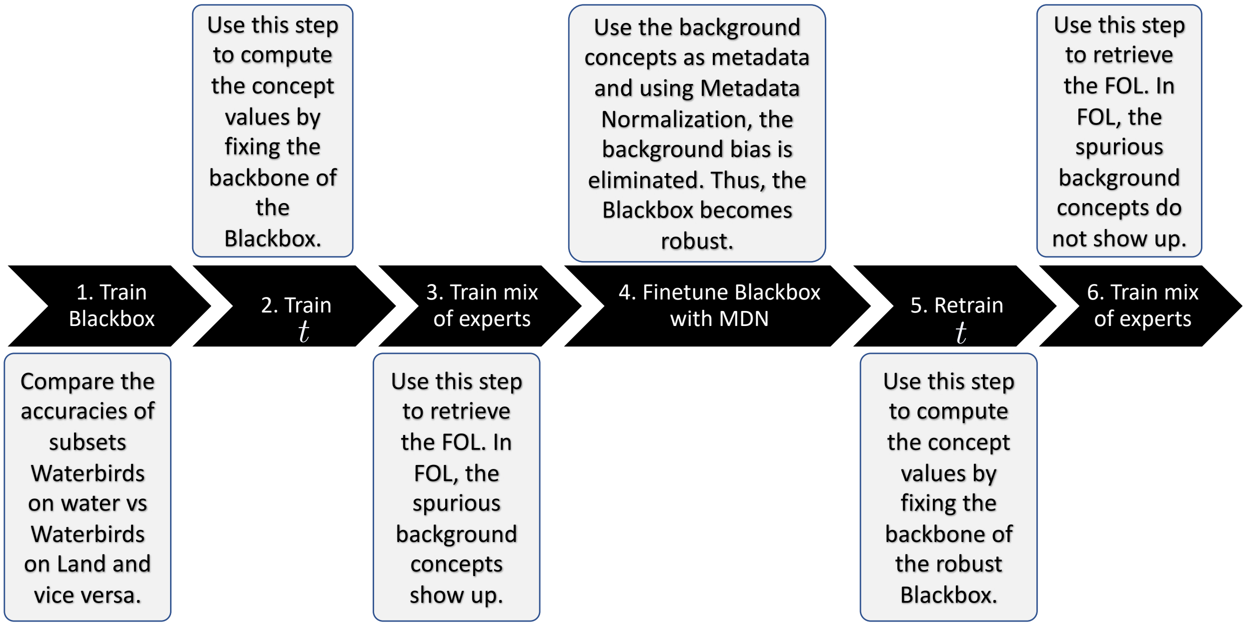

Use this step to compute the concept values by fixing the backbone of the Blackbox.
Use the background concepts as metadata and using Metadata Normalization, the background bias is eliminated. Thus, the Blackbox becomes robust.
Use this step to retrieve the FOL. In FOL, the spurious background concepts do not show up.
1. Train Blackbox
2. Train t
3. Train mix of experts
4. Finetune Blackbox with MDN
5. Retrain t
6. Train mix of experts
Compare the accuracies of subsets Waterbirds on water vs Waterbirds on Land and vice versa.
Use this step to retrieve the FOL. In FOL, the spurious background concepts show up.
Use this step to compute the concept values by fixing the backbone of the robust Blackbox.

Figure 8. The flow diagram to eliminate the shortcut from vision datasets using FOL by MoIE.

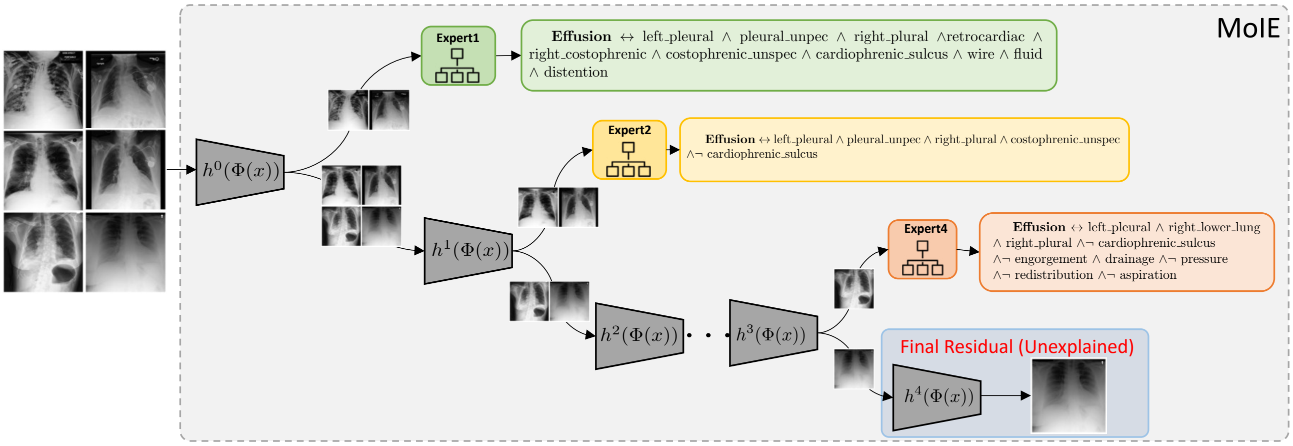

MoIE
Expert1
Effusion ↔ left_pleural ∧ pleural_unpec ∧ right_plural ∧retrocardiac ∧ right_costophrenic ∧ costophrenic_unspec ∧ cardiophrenic_sulcus ∧ wire ∧ fluid ∧ distention
Expert2
Effusion ↔ left_pleural ∧ pleural_unpec ∧ right_plural ∧ costophrenic_unspec ∧¬ cardiophrenic_sulcus
Expert4
Effusion ↔ left_pleural ∧ right_lower_lung ∧ right_plural ∧¬ cardiophrenic_sulcus ∧¬ engorgement ∧ drainage ∧¬ pressure ∧¬ redistribution ∧¬ aspiration
$h^0(\Phi(x))$
$h^1(\Phi(x))$
$h^2(\Phi(x))$
$h^3(\Phi(x))$
$h^4(\Phi(x))$
Final Residual (Unexplained)

Figure 9. Construction logical explanations of the samples of "Effusion" in the MIMIC-CXR dataset for various experts in MoIE at inference. The final residual covers the unexplained sample, which is "harder" to explain (indicated in *red*).

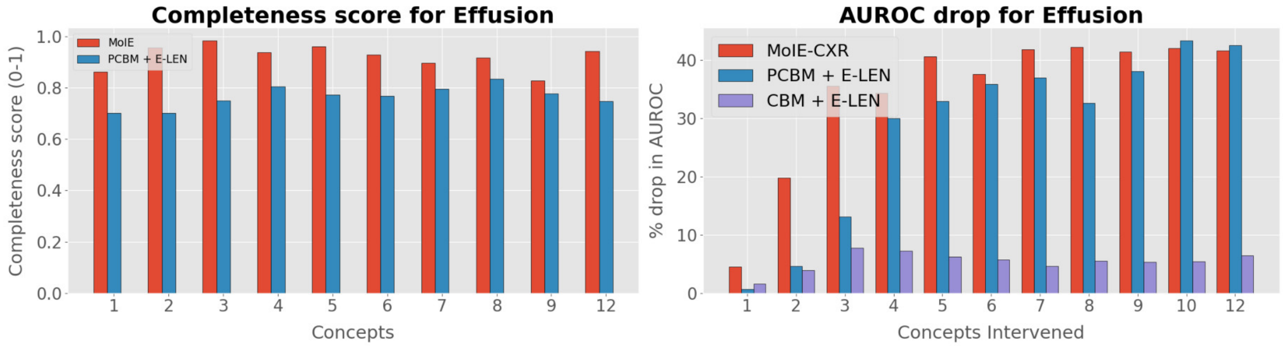


Figure 10. **(a)** Completeness scores for different significant concepts of Effusion. **(b)** Drop in AUROC by zeroing out the concepts for Effusion.

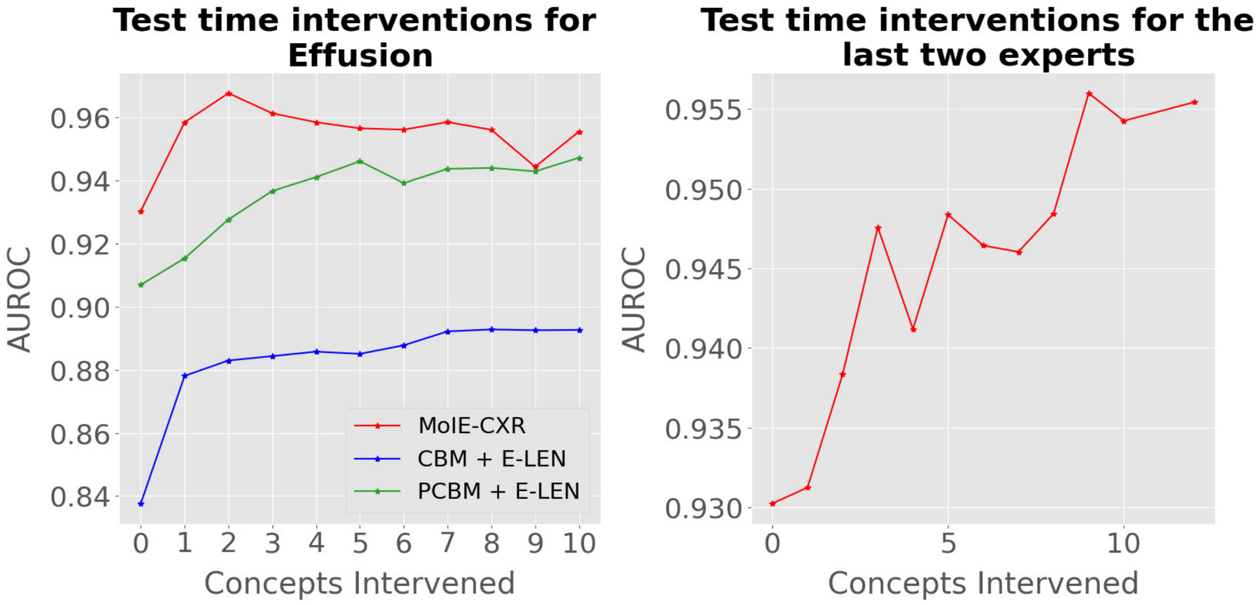
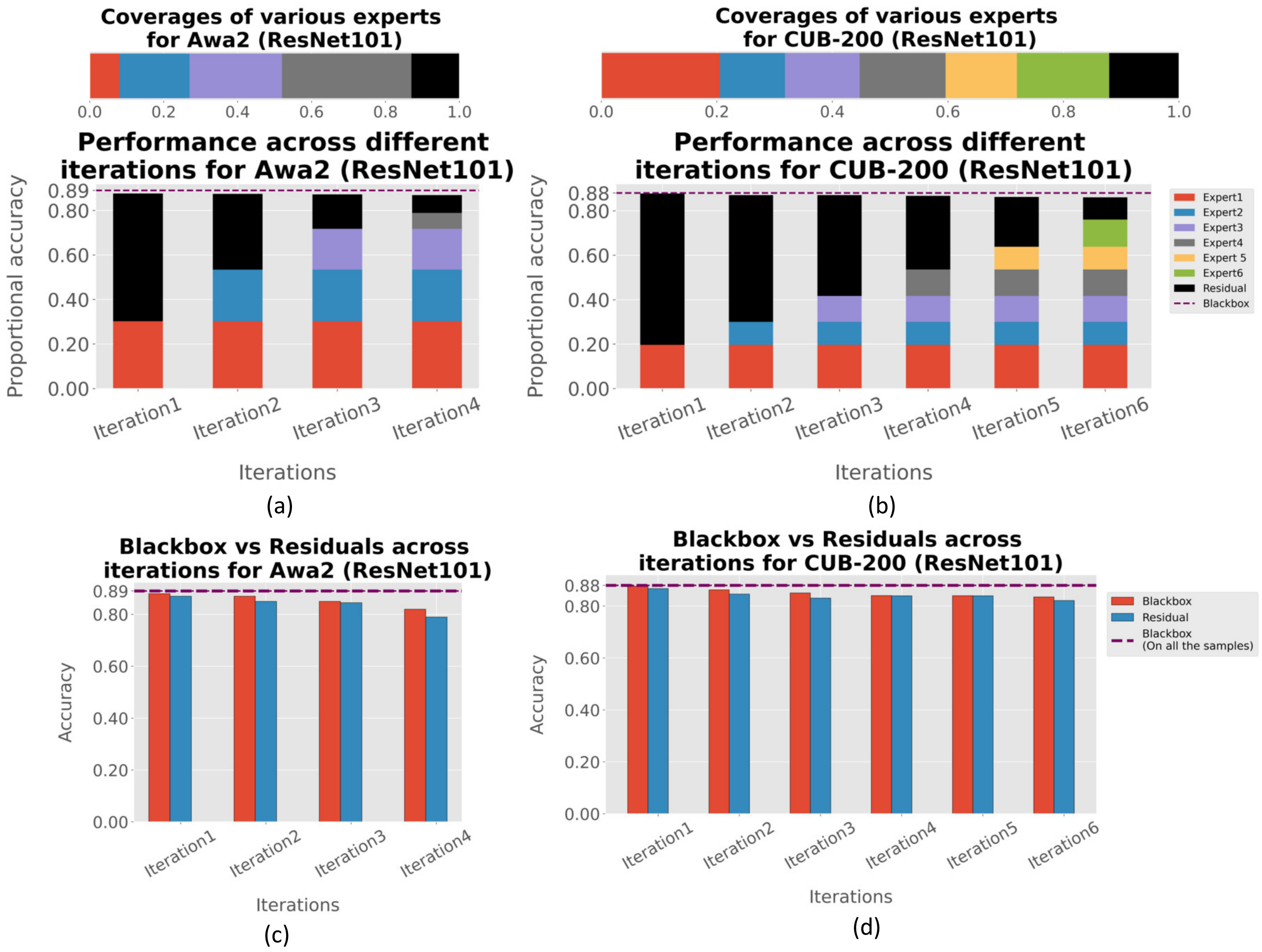


Figure 11. **(a)** Test time interventions of different concepts on all samples for Effusion. **(b)** Test time interventions of different concepts on only the “hard” samples covered by the last two experts for Effusion



(a)

(b)

(c)

(d)

Figure 12. The performances of experts and residuals across iterations for ResNet derived MoIE for CUB-200 and Awa2. **(a-b)** Coverage and proportional accuracy of the experts and residuals. **(c-d)** We route the samples covered by the residuals across iterations to the initial Blackbox f^0 and compare the accuracy of f^0 (red bar) with the residual (blue bar).

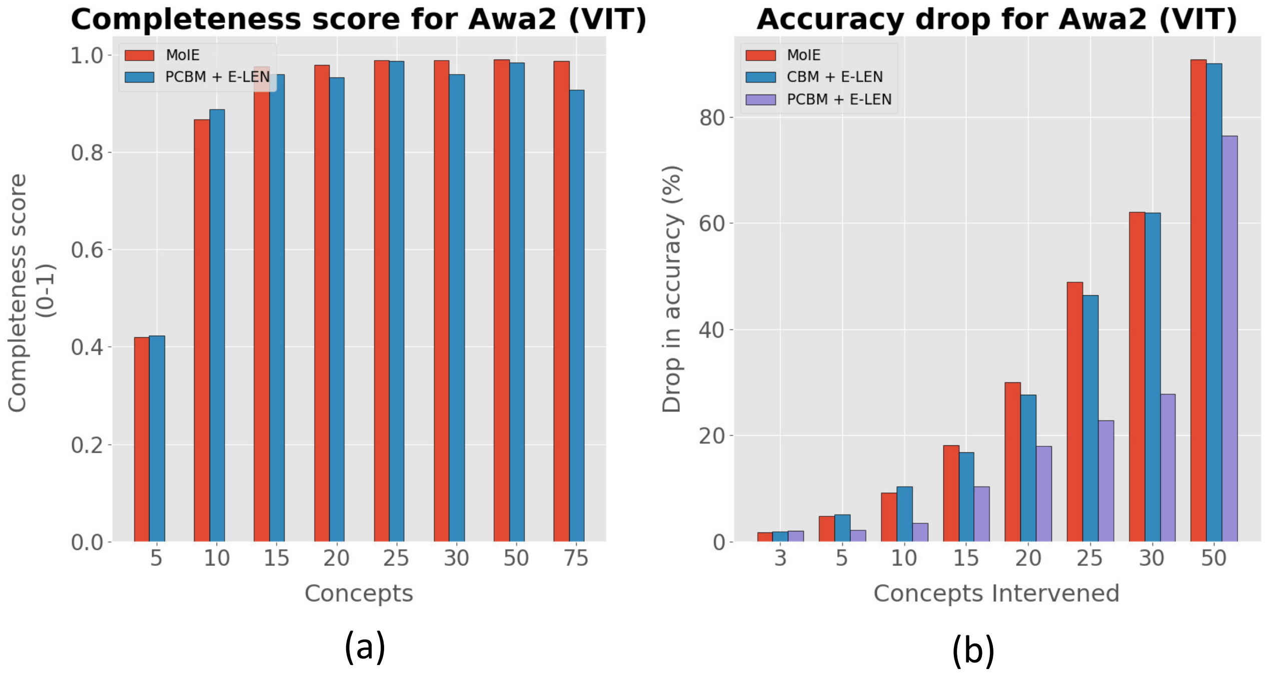


Figure 13. **(a)** Completeness scores for different significant concepts of Awa2. **(b)** Drop in accuracy by zeroing out the concepts for Awa2.

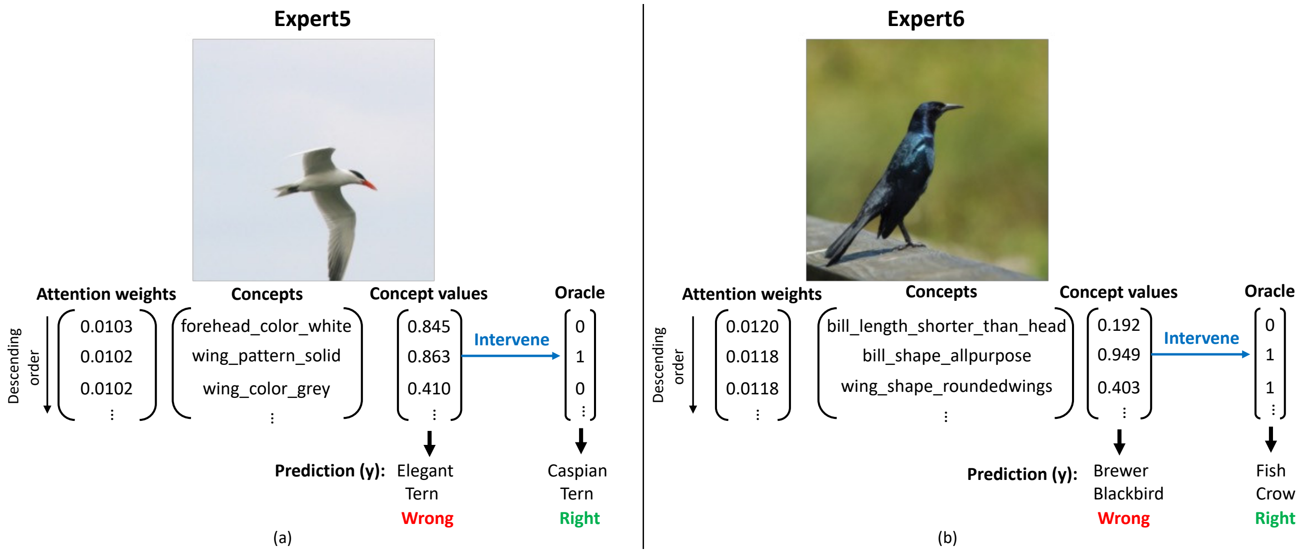


Figure 14. Illustration of test time intervention of top-3 concepts for "harder" samples identified by the last two experts of VIT-driven MoIE. We adopt E-LEN (Barbiero et al., 2022) as the experts. Thus, each concept is associated with an attention weight after training, signifying its prediction importance. So, here we intervene the top 3 concepts with the highest attention weights for samples routed to expert 5 (***left***) and expert 6 (***right***). These samples are considered the "harder" samples as they are routed to the last two experts of MoIE. We demonstrate that the test time intervention corrects the prediction.

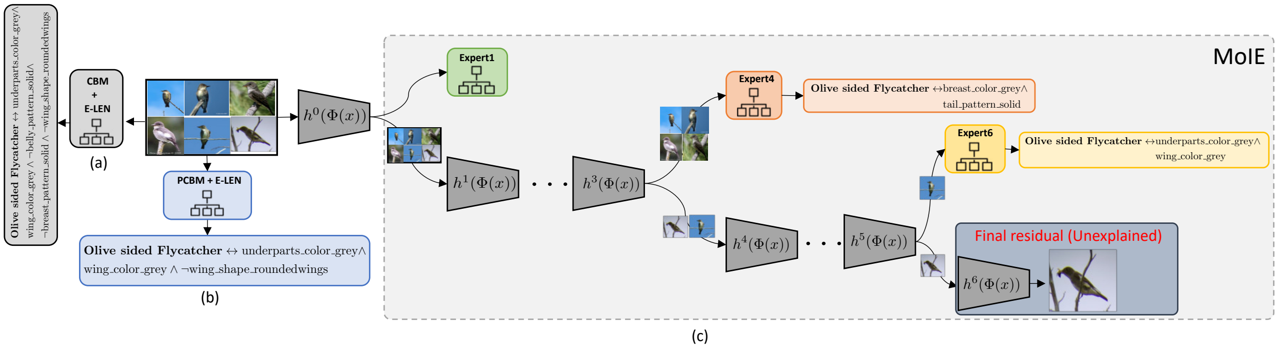

Olive sided Flycatcher ↔ underparts_color_grey∧
wing_color_grey ∧ ¬belly_pattern_solid∧
¬breast_pattern_solid ∧ ¬wing_shape_roundedwings
CBM
+
E-LEN
(a)
PCBM + E-LEN
Olive sided Flycatcher ↔ underparts_color_grey∧
wing_color_grey ∧ ¬wing_shape_roundedwings
(b)
$h^0(\Phi(x))$
MoIE
Expert1
$h^1(\Phi(x))$
$h^3(\Phi(x))$
Expert4
Olive sided Flycatcher ↔breast_color_grey∧
tail_pattern_solid
Expert6
Olive sided Flycatcher ↔underparts_color_grey∧
wing_color_grey
$h^4(\Phi(x))$
$h^5(\Phi(x))$
Final residual (Unexplained)
$h^6(\Phi(x))$
(c)

Figure 15. Construction logical explanations of the samples of a category, “Olive sided Flycatcher” in the CUB-200 dataset for (a) VIT-based sequential CBM + E-LEN as an *interpretable by design* baseline, (b) VIT-based PCBM + E-LEN as a posthoc based baseline, (c) various experts in MoIE at inference. This is an example where the final residual covers the unexplained sample, which is “harder” to explain (indicated in *red*). Also, MoIE can capture more instance-specific concepts than generic ones by the baselines.

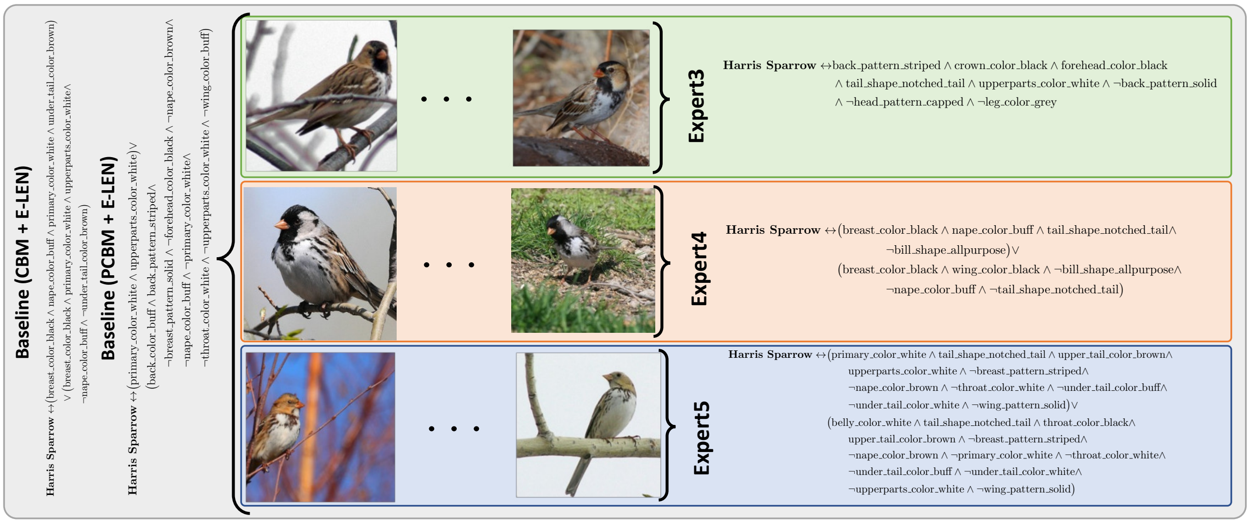

Baseline (CBM + E-LEN)
Harris Sparrow ↔(breast_color_black ∧ nape_color_buff ∧ primary_color_white ∧ under_tail_color_brown)
∨ (breast_color_black ∧ primary_color_white ∧ upperparts_color_white∧
¬nape_color_buff ∧ ¬under_tail_color_brown)
Baseline (PCBM + E-LEN)
Harris Sparrow ↔(primary_color_white ∧ upperparts_color_white)∨
(back_color_buff ∧ back_pattern_striped∧
¬breast_pattern_solid ∧ ¬forehead_color_black ∧ ¬nape_color_brown∧
¬nape_color_buff ∧ ¬primary_color_white∧
¬throat_color_white ∧ ¬upperparts_color_white ∧ ¬wing_color_buff)
Expert3
Harris Sparrow ↔back_pattern_striped ∧ crown_color_black ∧ forehead_color_black
∧ tail_shape_notched_tail ∧ upperparts_color_white ∧ ¬back_pattern_solid
∧ ¬head_pattern_capped ∧ ¬leg_color_grey
Expert4
Harris Sparrow ↔(breast_color_black ∧ nape_color_buff ∧ tail_shape_notched_tail∧
¬bill_shape_allpurpose)∨
(breast_color_black ∧ wing_color_black ∧ ¬bill_shape_allpurpose∧
¬nape_color_buff ∧ ¬tail_shape_notched_tail)
Expert5
Harris Sparrow ↔(primary_color_white ∧ tail_shape_notched_tail ∧ upper_tail_color_brown∧
upperparts_color_white ∧ ¬breast_pattern_striped∧
¬nape_color_brown ∧ ¬throat_color_white ∧ ¬under_tail_color_buff∧
¬under_tail_color_white ∧ ¬wing_pattern_solid)∨
(belly_color_white ∧ tail_shape_notched_tail ∧ throat_color_black∧
upper_tail_color_brown ∧ ¬breast_pattern_striped∧
¬nape_color_brown ∧ ¬primary_color_white ∧ ¬throat_color_white∧
¬under_tail_color_buff ∧ ¬under_tail_color_white∧
¬upperparts_color_white ∧ ¬wing_pattern_solid)

Figure 16. Construction logical explanations of the samples of a category, “Harris Sparrow” in the CUB-200 dataset for (a) VIT-based sequential CBM + E-LEN as an *interpretable by design* baseline, (b) VIT-based PCBM + E-LEN as a posthoc based baseline, (c) various experts in MoIE at inference.

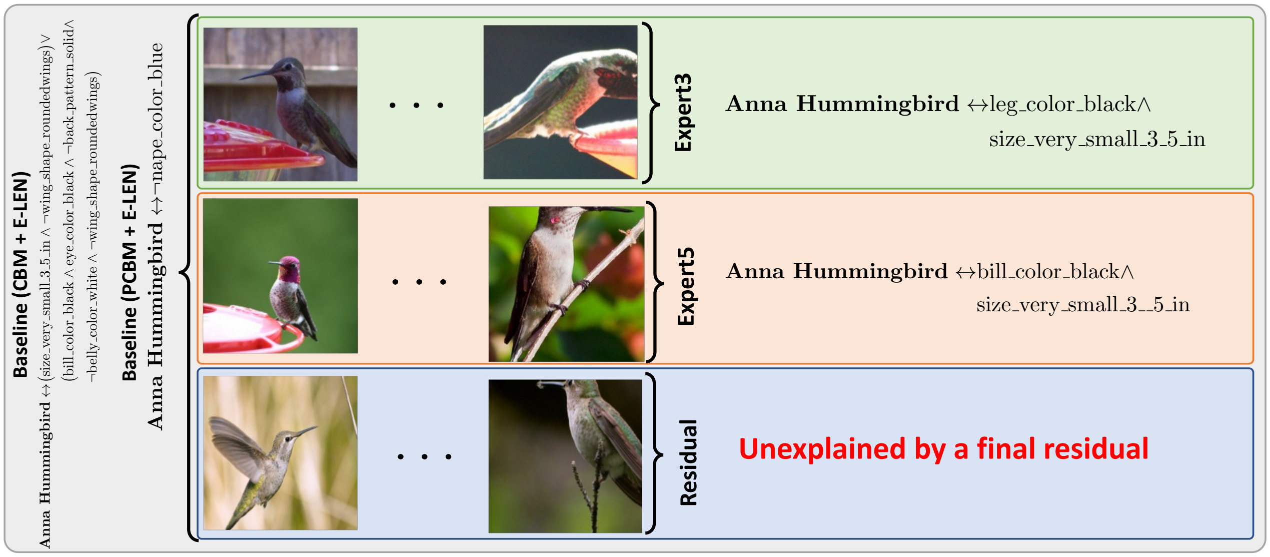

Baseline (CBM + E-LEN)
Anna Hummingbird ↔(size_very_small_3_5_in ∧ ¬wing_shape_roundedwings) ∨
(bill_color_black ∧ eye_color_black ∧ ¬back_pattern_solid∧
¬belly_color_white ∧ ¬wing_shape_roundedwings)
Baseline (PCBM + E-LEN)
Anna Hummingbird ↔¬nape_color_blue
Expert3
Anna Hummingbird ↔leg_color_black∧
size_very_small_3_5_in
Expert5
Anna Hummingbird ↔bill_color_black∧
size_very_small_3__5_in
Residual
Unexplained by a final residual

Figure 17. Construction logical explanations of the samples of a category, "Anna Hummingbird" in the CUB-200 dataset for (a) VIT-based sequential CBM + E-LEN as an *interpretable by design* baseline, (b) VIT-based PCBM + E-LEN as a posthoc based baseline, (c) various experts in MoIE at inference.

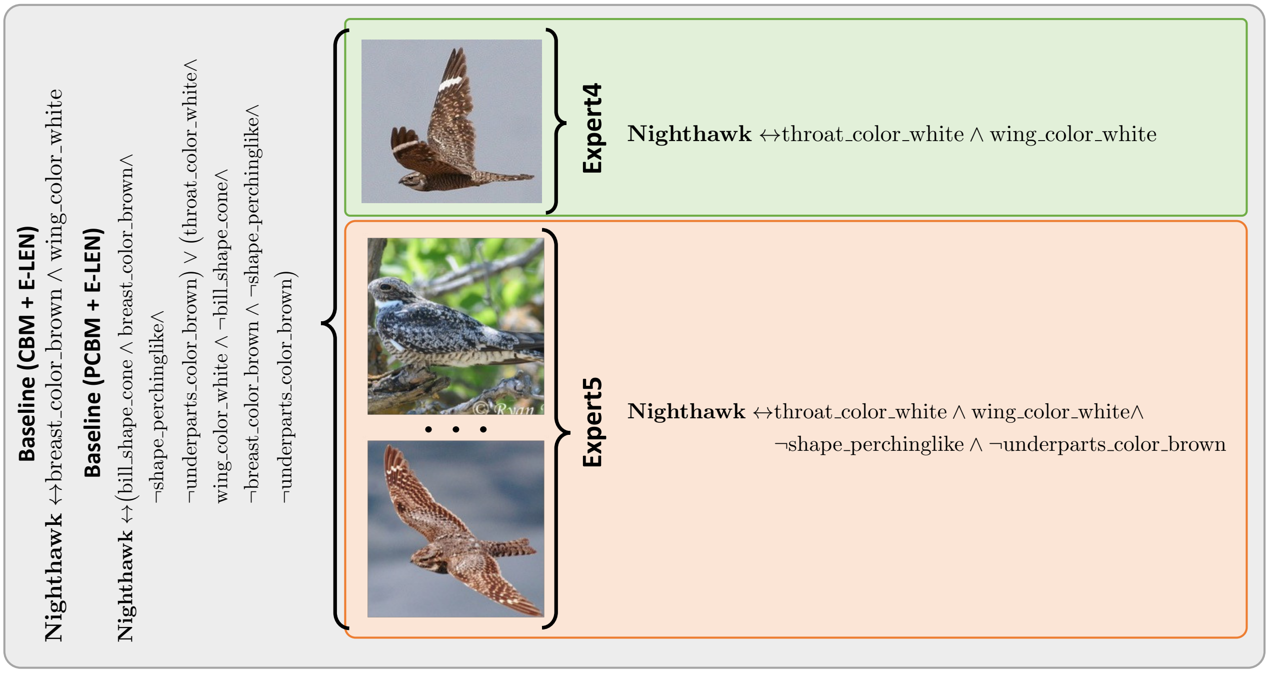

Baseline (CBM + E-LEN)
Nighthawk ↔breast_color_brown ∧ wing_color_white
Baseline (PCBM + E-LEN)
Nighthawk ↔(bill_shape_cone ∧ breast_color_brown∧
¬shape_perchinglike∧
¬underparts_color_brown) ∨ (throat_color_white∧
wing_color_white ∧ ¬bill_shape_cone∧
¬breast_color_brown ∧ ¬shape_perchinglike∧
¬underparts_color_brown)
Expert4
Nighthawk ↔throat_color_white ∧ wing_color_white
Expert5
Nighthawk ↔throat_color_white ∧ wing_color_white∧
¬shape_perchinglike ∧ ¬underparts_color_brown

Figure 18. Construction logical explanations of the samples of a category, "Nighthawk" in the CUB-200 dataset for (a) VIT-based sequential CBM + E-LEN as an *interpretable by design* baseline, (b) VIT-based PCBM + E-LEN as a posthoc based baseline, (c) various experts in MoIE at inference.

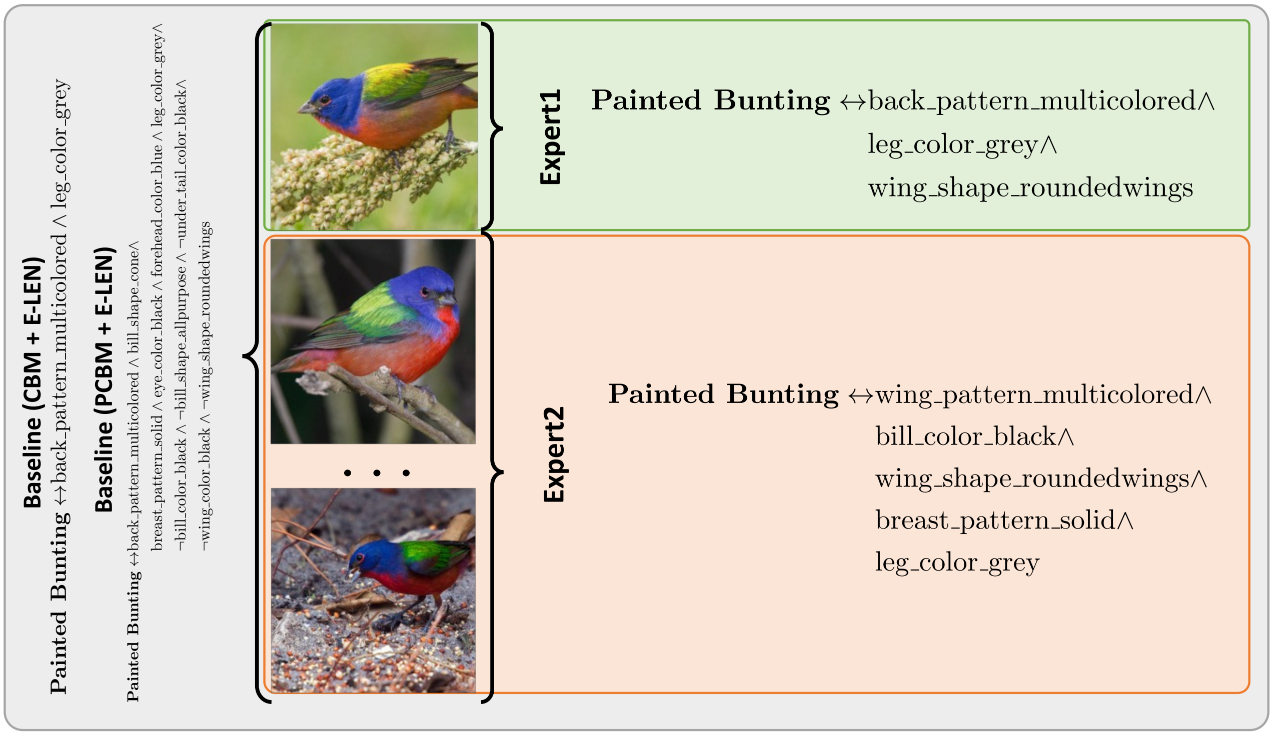

Baseline (CBM + E-LEN)
Painted Bunting ↔back_pattern_multicolored ∧ leg_color_grey
Baseline (PCBM + E-LEN)
Painted Bunting ↔back_pattern_multicolored ∧ bill_shape_cone∧
breast_pattern_solid ∧ eye_color_black ∧ forehead_color_blue ∧ leg_color_grey∧
¬bill_color_black ∧ ¬bill_shape_allpurpose ∧ ¬under_tail_color_black∧
¬wing_color_black ∧ ¬wing_shape_roundedwings
Expert1
Painted Bunting ↔back_pattern_multicolored∧
leg_color_grey∧
wing_shape_roundedwings
Expert2
Painted Bunting ↔wing_pattern_multicolored∧
bill_color_black∧
wing_shape_roundedwings∧
breast_pattern_solid∧
leg_color_grey

Figure 19. Construction logical explanations of the samples of a category, "Painted Bunting" in the CUB-200 dataset for (a) VIT-based sequential CBM + E-LEN as an *interpretable by design* baseline, (b) VIT-based PCBM + E-LEN as a posthoc based baseline, (c) various experts in MoIE at inference.

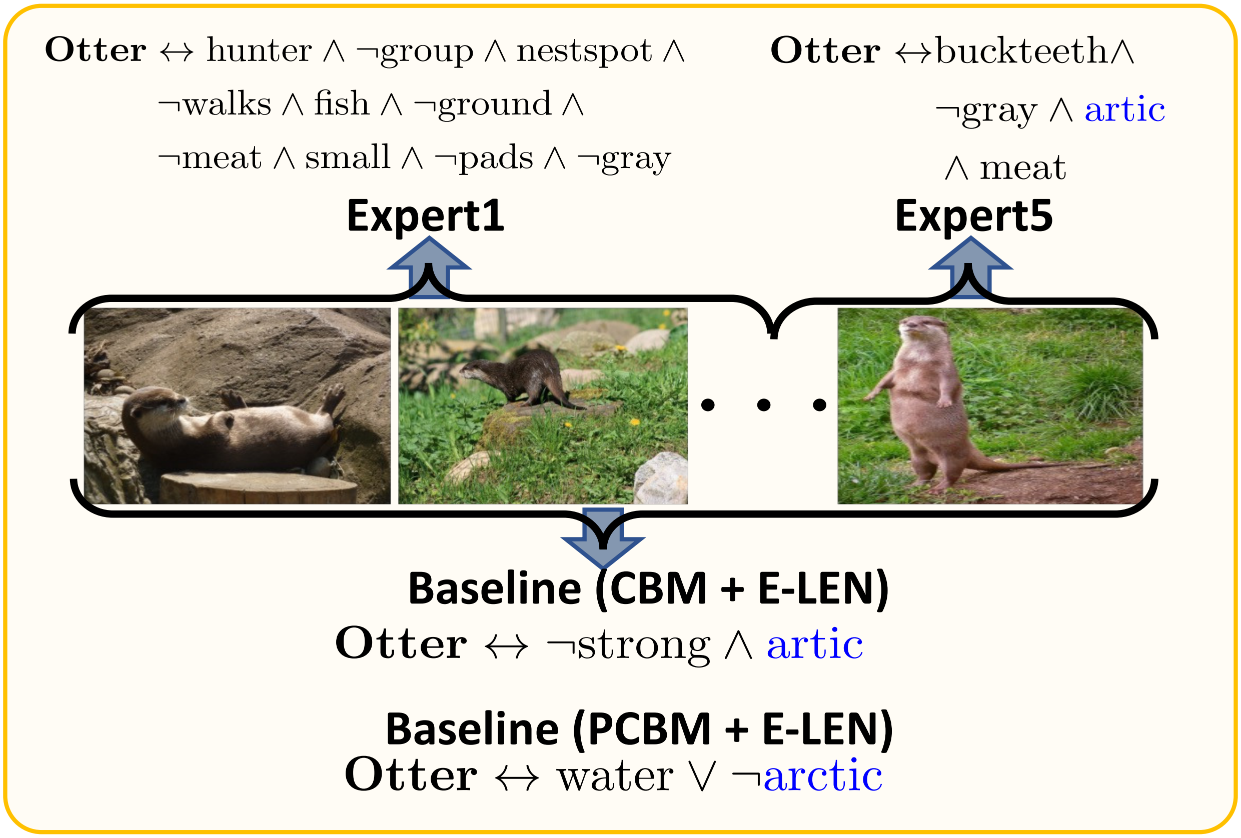
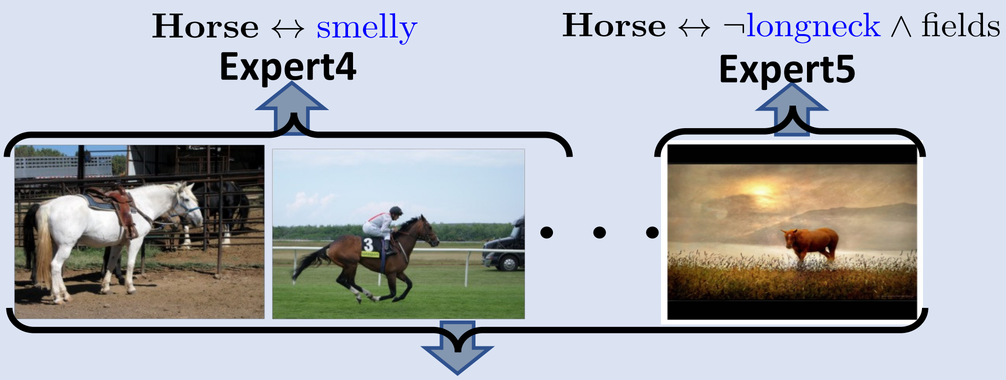


Figure 20. Flexibility of FOL explanations by VIT-derived MoIE MoIE and the CBM + E-LEN and PCBM + E-LEN baselines for Awa2 dataset to classify "Otter" at inference. Both the baseline's FOL constitutes identical concepts to distinguish all the samples. However, expert1 classifies "Otter" with *hunter*, *group etc.*as the identifying concept for the instances covered by it. Similarly expert5 classifies "Otter" using *buckteeth*, *gray etc.*. Note that, *meat* and *gray* are shared between the two experts. We highlight the shared concepts (*artic*) between the experts and the baselines as blue.



$\textbf{Horse} \leftrightarrow \text{smelly}$
Expert4
$\textbf{Horse} \leftrightarrow \neg\text{longneck} \wedge \text{fields}$
Expert5

Baseline (CBM + E-LEN)

$\textbf{Horse} \leftrightarrow (\text{buckteeth} \wedge \text{longneck}) \vee (\text{longneck} \wedge \text{smelly}) \vee (\text{longleg} \wedge \text{smelly} \wedge \neg\text{buckteeth})$

Baseline (PCBM + E-LEN)

$$
\begin{aligned}
\textbf{Horse} \leftrightarrow\ & (\text{buckteeth} \wedge \text{longneck}) \vee (\neg\text{buckteeth} \wedge \neg\text{longneck}) \vee \\
& (\text{buckteeth} \wedge \text{bulbous} \wedge \text{gray} \wedge \text{longleg} \wedge \text{longneck} \wedge \neg\text{forager} \wedge \neg\text{solitary} \wedge \neg\text{spots}) \vee \\
& (\text{buckteeth} \wedge \text{gray} \wedge \text{longleg} \wedge \text{longneck} \wedge \neg\text{bulbous} \wedge \neg\text{forager} \wedge \neg\text{solitary} \wedge \neg\text{spots}) \vee \\
& (\\
& \quad \text{active} \wedge \text{buckteeth} \wedge \text{chewteeth} \wedge \text{hooves} \wedge \text{horns} \wedge \text{lean} \wedge \text{longleg} \wedge \text{longneck} \wedge \\
& \quad \text{muscle} \wedge \text{oldworld} \wedge \text{patches} \wedge \text{smelly} \wedge \text{tail} \wedge \text{timid} \wedge \text{toughskin} \wedge \neg\text{bulbous} \wedge \\
& \quad \neg\text{bush} \wedge \neg\text{forager} \wedge \neg\text{forest} \wedge \neg\text{gray} \wedge \neg\text{hairless} \wedge \neg\text{inactive} \wedge \neg\text{meatteeth} \wedge \\
& \quad \neg\text{mountains} \wedge \neg\text{nestspot} \wedge \neg\text{paws} \wedge \neg\text{small} \wedge \neg\text{solitary} \wedge \neg\text{spots} \\
&) \vee \\
& (\\
& \quad \text{active} \wedge \text{big} \wedge \text{black} \wedge \text{bulbous} \wedge \text{chewteeth} \wedge \text{furry} \wedge \text{grazer} \wedge \text{ground} \wedge \\
& \quad \text{hooves} \wedge \text{horns} \wedge \text{inactive} \wedge \text{longleg} \wedge \text{longneck} \wedge \text{muscle} \wedge \text{oldworld} \wedge \\
& \quad \text{patches} \wedge \text{quadrapedal} \wedge \text{slow} \wedge \text{smelly} \wedge \text{strong} \wedge \text{tail} \wedge \text{timid} \wedge \\
& \quad \text{toughskin} \wedge \text{walks} \wedge \text{white} \wedge \neg\text{agility} \wedge \neg\text{arctic} \wedge \neg\text{buckteeth} \wedge \neg\text{bush} \wedge \\
& \quad \neg\text{claws} \wedge \neg\text{coastal} \wedge \neg\text{fast} \wedge \neg\text{fierce} \wedge \neg\text{fish} \wedge \neg\text{flippers} \wedge \neg\text{forager} \wedge \\
& \quad \neg\text{forest} \wedge \neg\text{gray} \wedge \neg\text{hairless} \wedge \neg\text{hibernate} \wedge \neg\text{hunter} \wedge \neg\text{jungle} \wedge \\
& \quad \neg\text{lean} \wedge \neg\text{meat} \wedge \neg\text{meatteeth} \wedge \neg\text{mountains} \wedge \neg\text{nestspot} \wedge \neg\text{nocturnal} \wedge \\
& \quad \neg\text{ocean} \wedge \neg\text{pads} \wedge \neg\text{paws} \wedge \neg\text{small} \wedge \neg\text{smart} \wedge \neg\text{solitary} \wedge \neg\text{spots} \wedge \\
& \quad \neg\text{stripes} \wedge \neg\text{swims} \wedge \neg\text{tunnels} \wedge \neg\text{water} \wedge \neg\text{weak} \\
&)
\end{aligned}
$$

Figure 21. Flexibility of FOL explanations by VIT-derived MoIE MoIE and the CBM + E-LEN and PCBM + E-LEN baselines for Awa2 dataset to classify "Horse" at inference. Both the baseline's FOL constitutes identical concepts to distinguish all the samples. However, expert4 classifies "Horse" with *smelly* as the identifying concept for the instances covered by it. Similarly, expert5 classifies the same "Horse" using *longneck* and *fields*. We highlight the shared concepts between the experts and the baselines as blue.

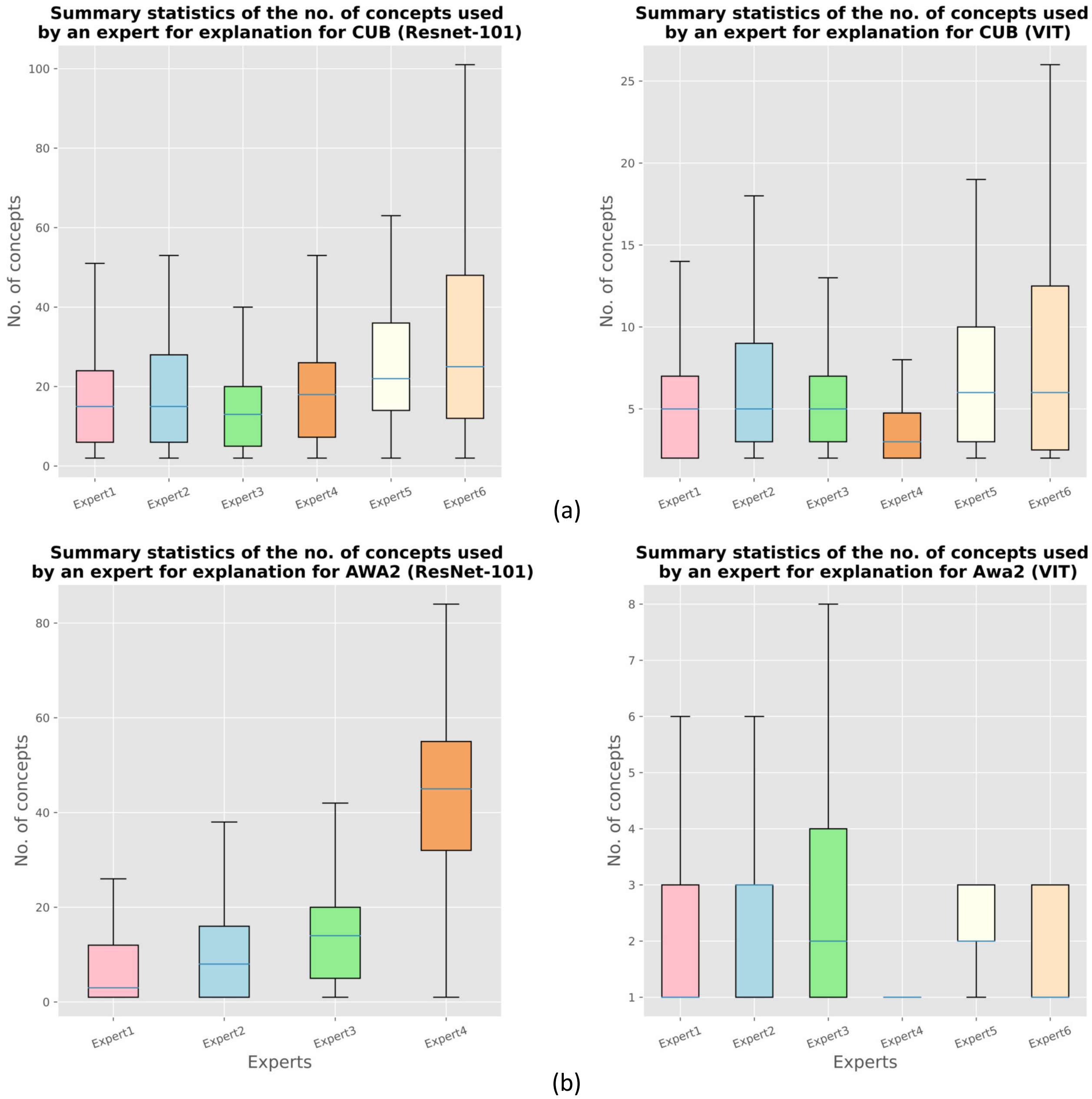
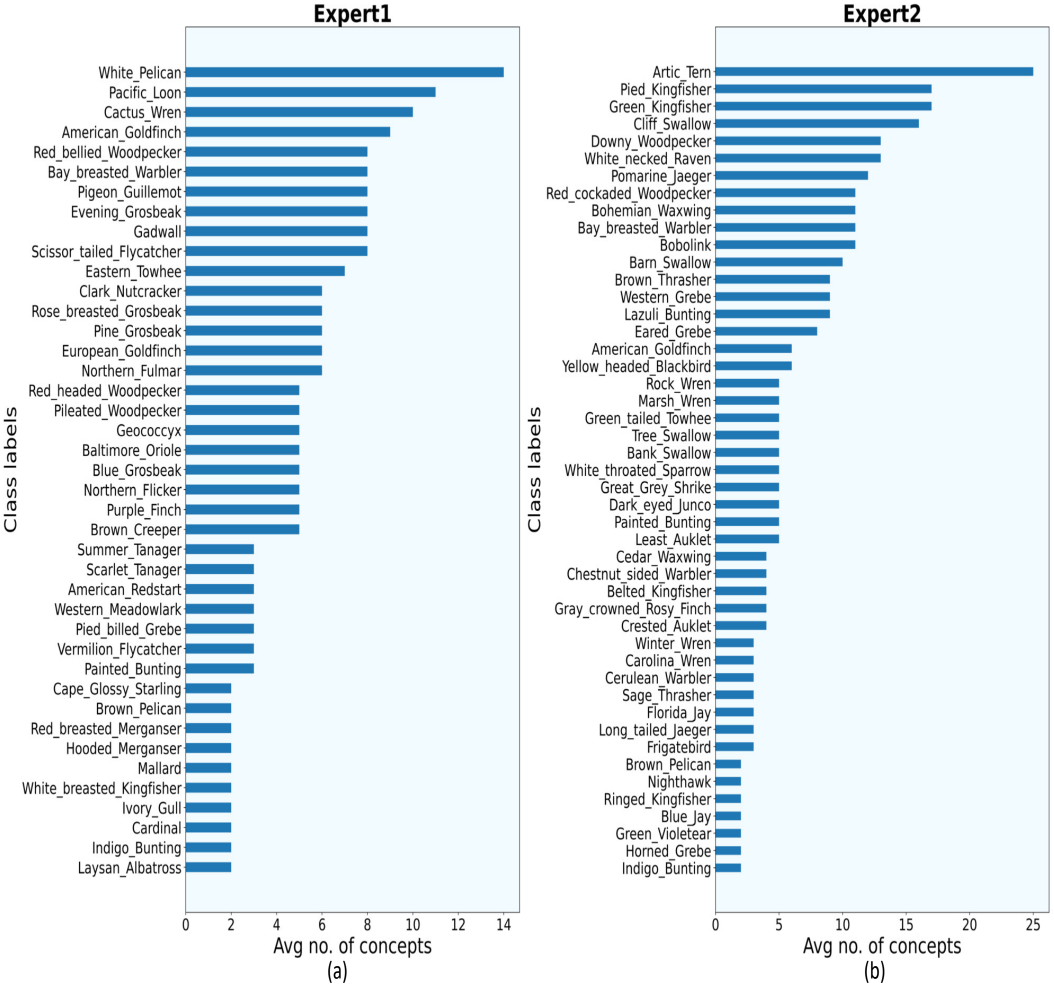


Figure 22. Summary statistics of the number of concepts utilized by various experts of datasets (a) CUB -200(top row) and (b) Awa2 (bottom row). In general, we can see that experts carving out the explanations from VIT often uses less number of concepts.



(a)

(b)

Figure 23. Class labels (Bird species) vs. avg concepts using VIT as the backbone for CUB-200 by (a) Expert1 (b) Expert2. Each bar in this plot indicates the average number of concepts required to explain each sample of that bird species correctly. For example according to (a) expert1 requires 14 concepts to explain an instance of "White Pelican".

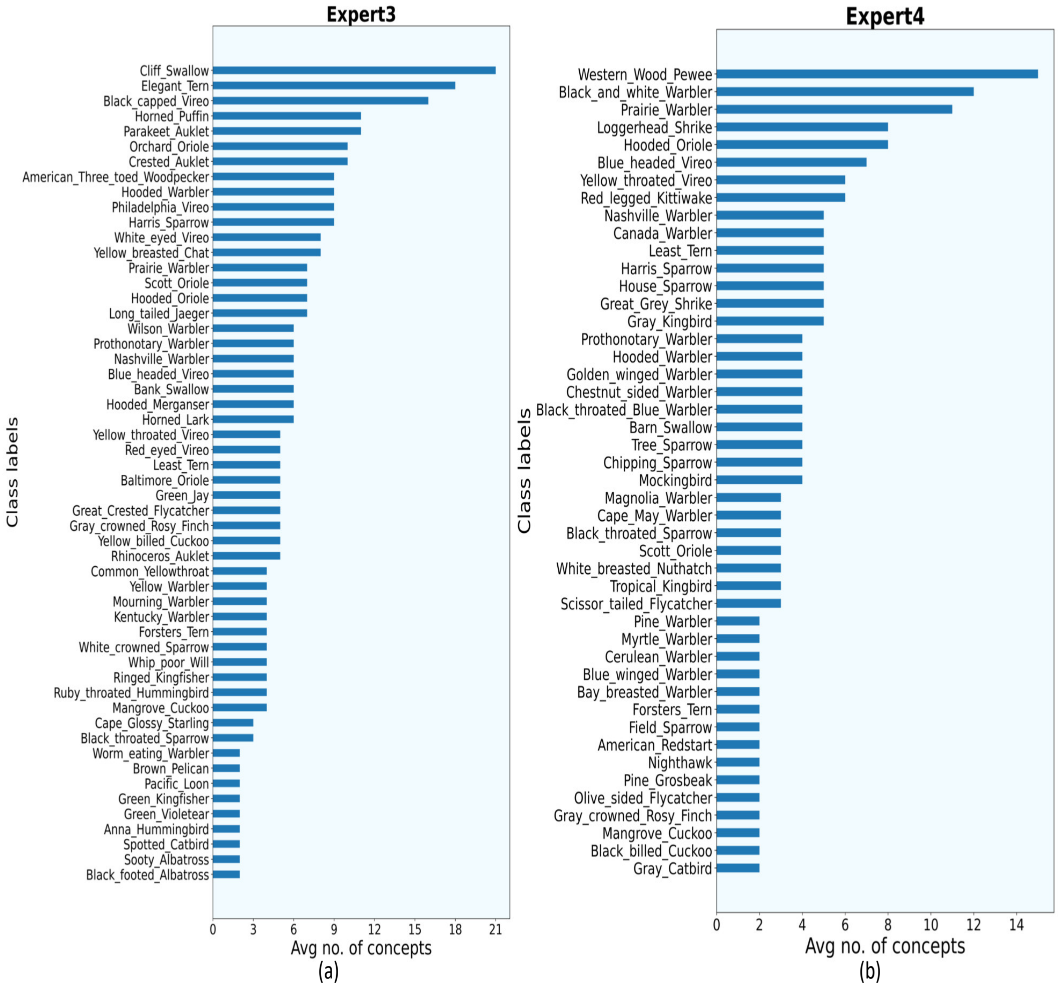


Figure 24. Class labels (Bird species) vs. avg concepts using VIT as the backbone for CUB-200 by (a) Expert3 (b) Expert4. Each bar in this plot indicates the average number of concepts required to explain each sample of that bird species correctly. For example according to (a) expert3 requires 21 concepts to explain an instance of "Cliff Swallow".

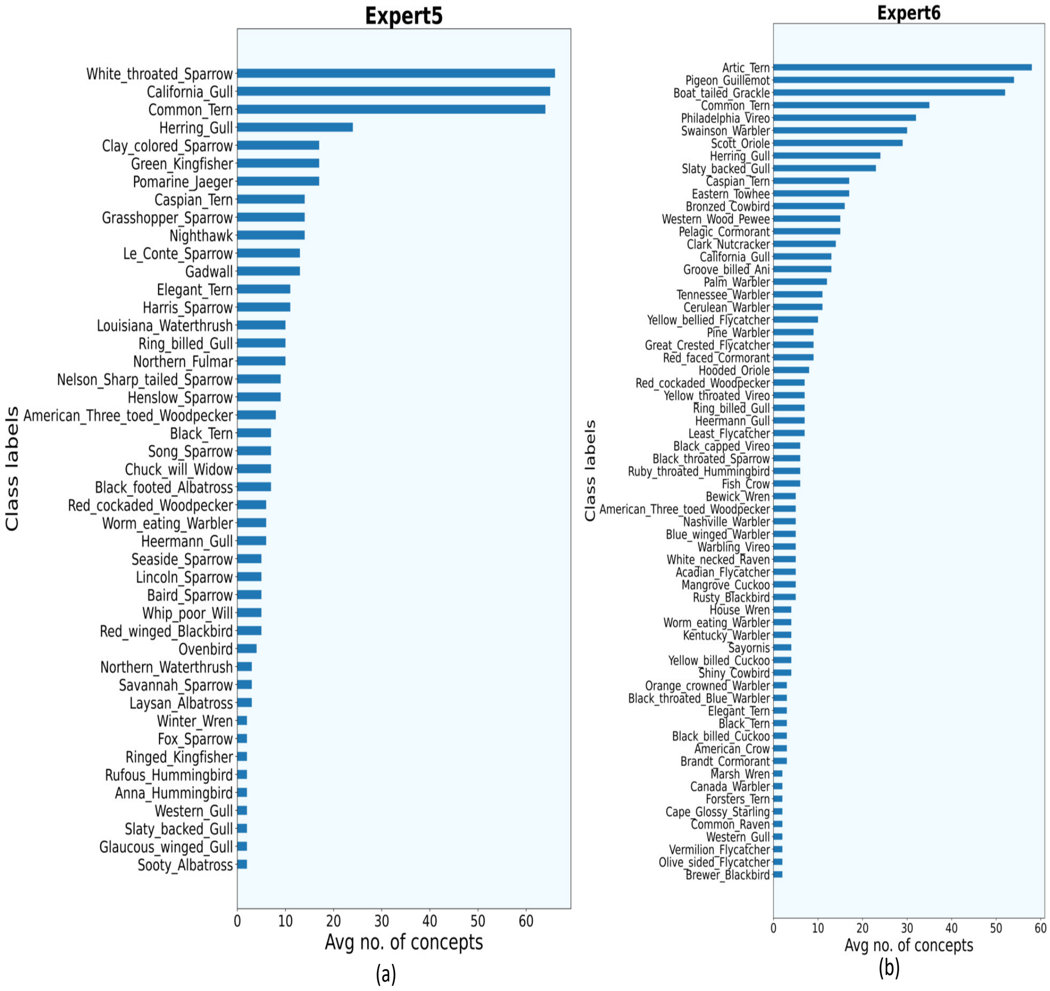


Figure 25. Class labels (Bird species) vs. avg concepts using VIT as the backbone for CUB-200 by (a) Expert5 (b) Expert6. Each bar in this plot indicates the average number of concepts required to explain each sample of that bird species correctly. For example according to (a) expert5 requires approximately 65 concepts to explain an instance of "White throated sparrow".

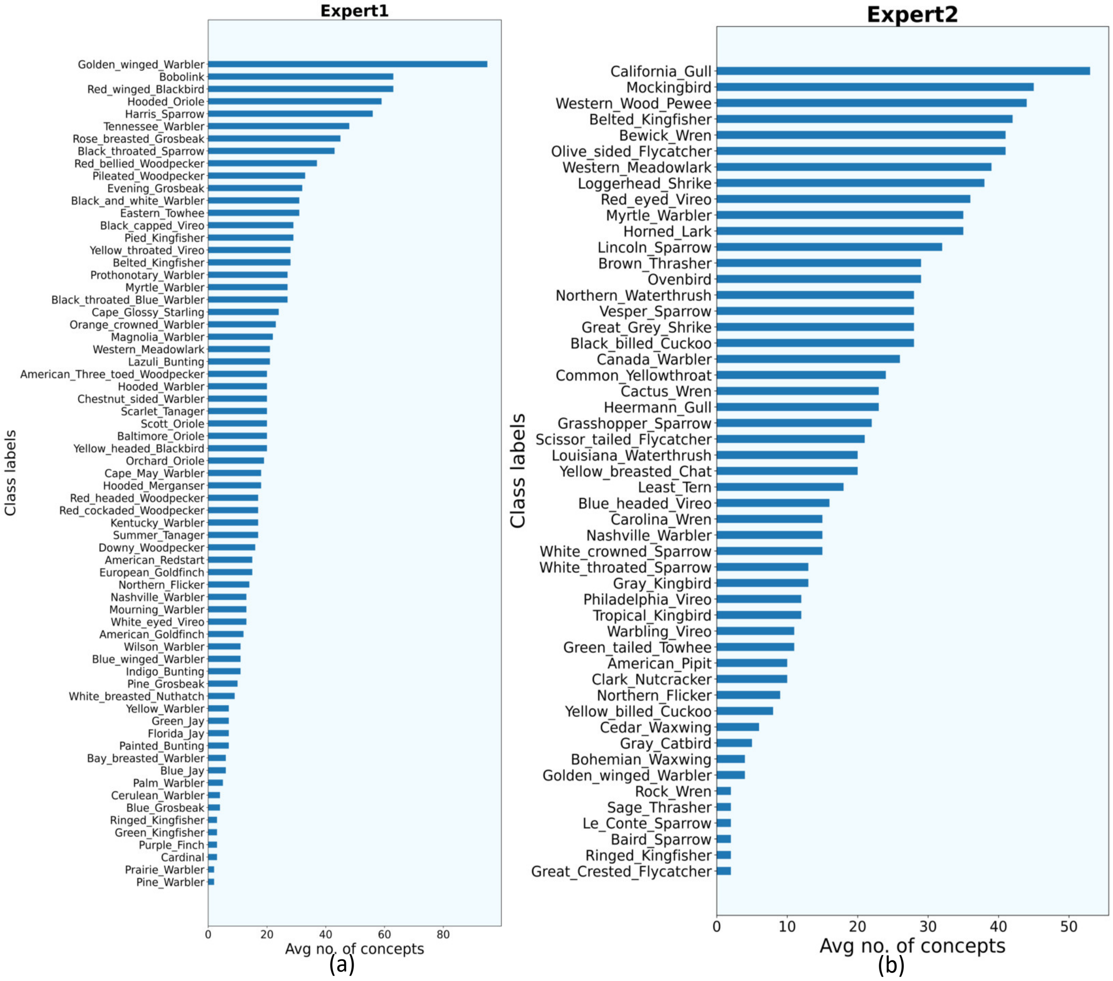


Figure 26. Class labels (Bird species) vs. avg concepts using ResNet-101 as the backbone for CUB-200 by (a) Expert1 (b) Expert2. Each bar in this plot indicates the average number of concepts required to explain each sample of that bird species correctly. For example according to (a) expert1 requires approximately 85 concepts to explain an instance of "Golden winged warbler".

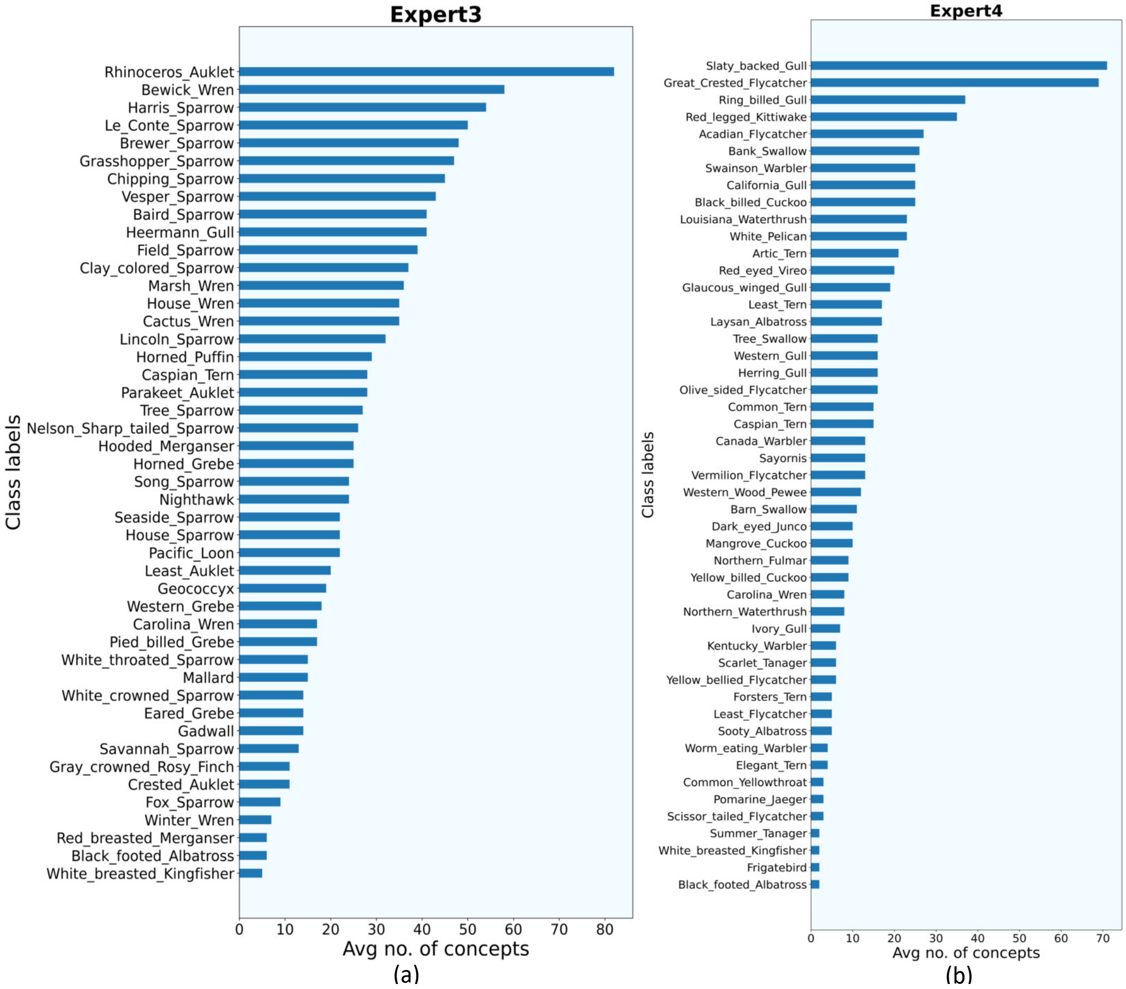
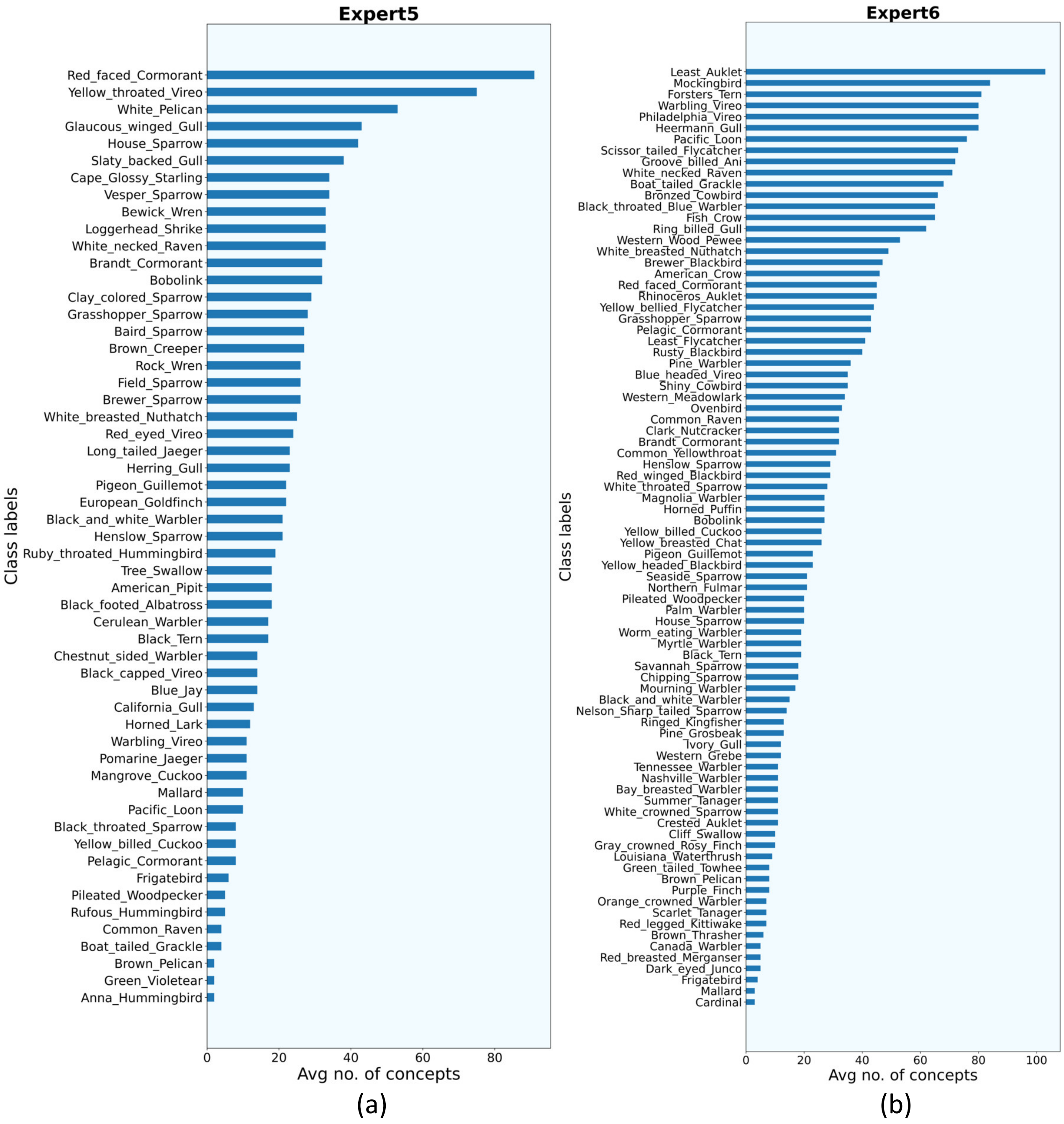


Figure 27. Class labels (Bird species) vs. avg concepts using ResNet-101 as the backbone for CUB-200 by (a) Expert3 (b) Expert4. Each bar in this plot indicates the average number of concepts required to explain each sample of that bird species correctly. For example according to (a) expert3 requires approximately 82 concepts to explain an instance of "Rhinoceros auklet".



(a)

(b)

Figure 28. Class labels (Bird species) vs. avg concepts using ResNet-101 as the backbone for CUB-200 by (a) Expert5 (b) Expert6. Each bar in this plot indicates the average number of concepts required to explain each sample of that bird species correctly. For example according to (a) expert5 requires approximately 85 concepts to explain an instance of "Red faced carmorant".

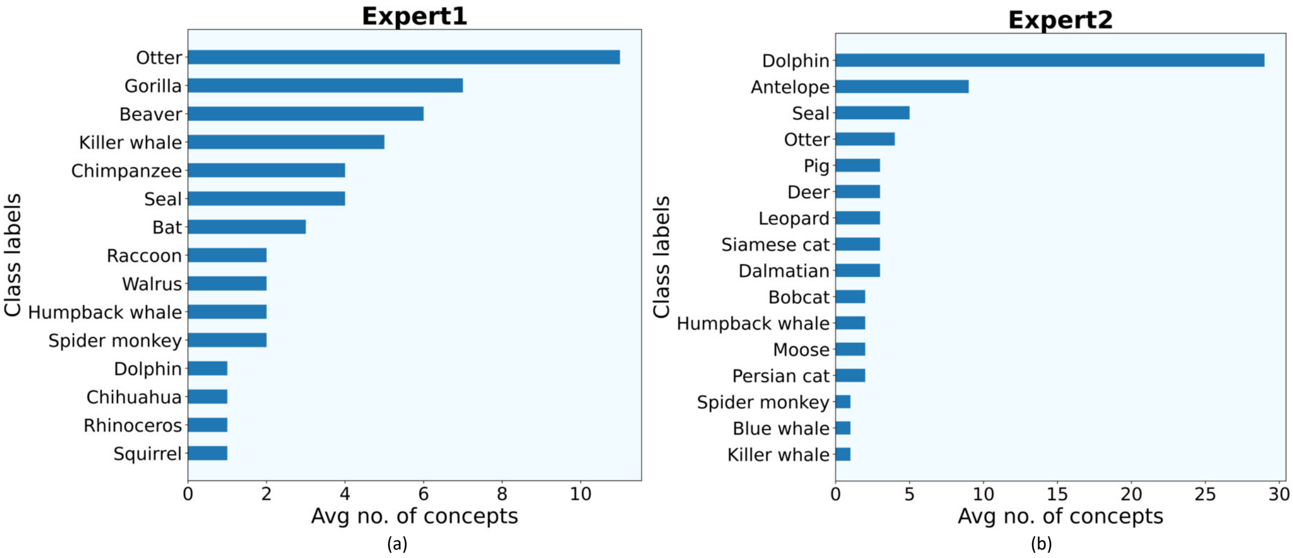


Figure 29. Class labels (Animal species) vs. avg concepts using VIT as the backbone for Awa2. Each bar in this plot indicates the average number of concepts required to explain each sample of that animal species correctly. For example according to (c) expert1 requires approximately 12 concepts to explain an instance of “Otter”.

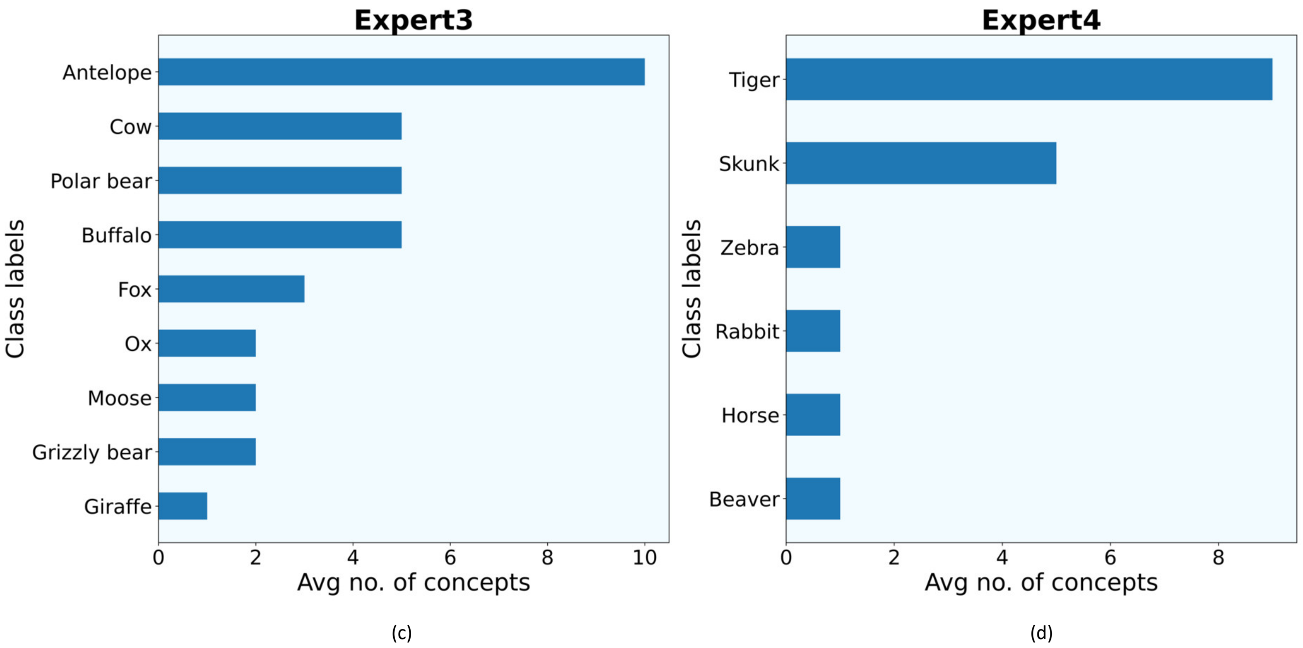
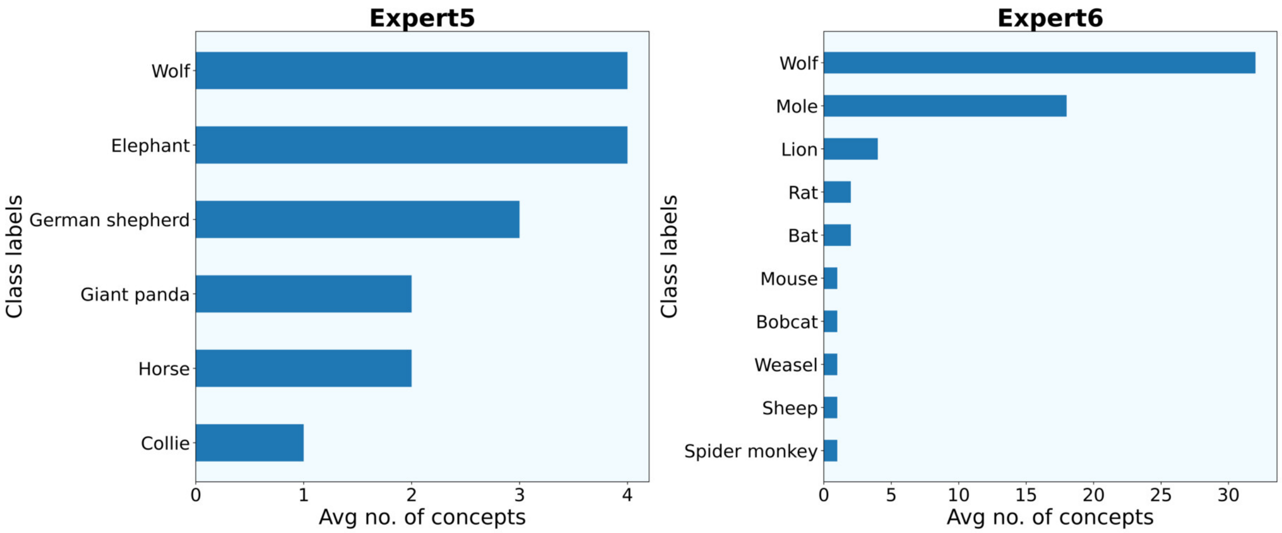


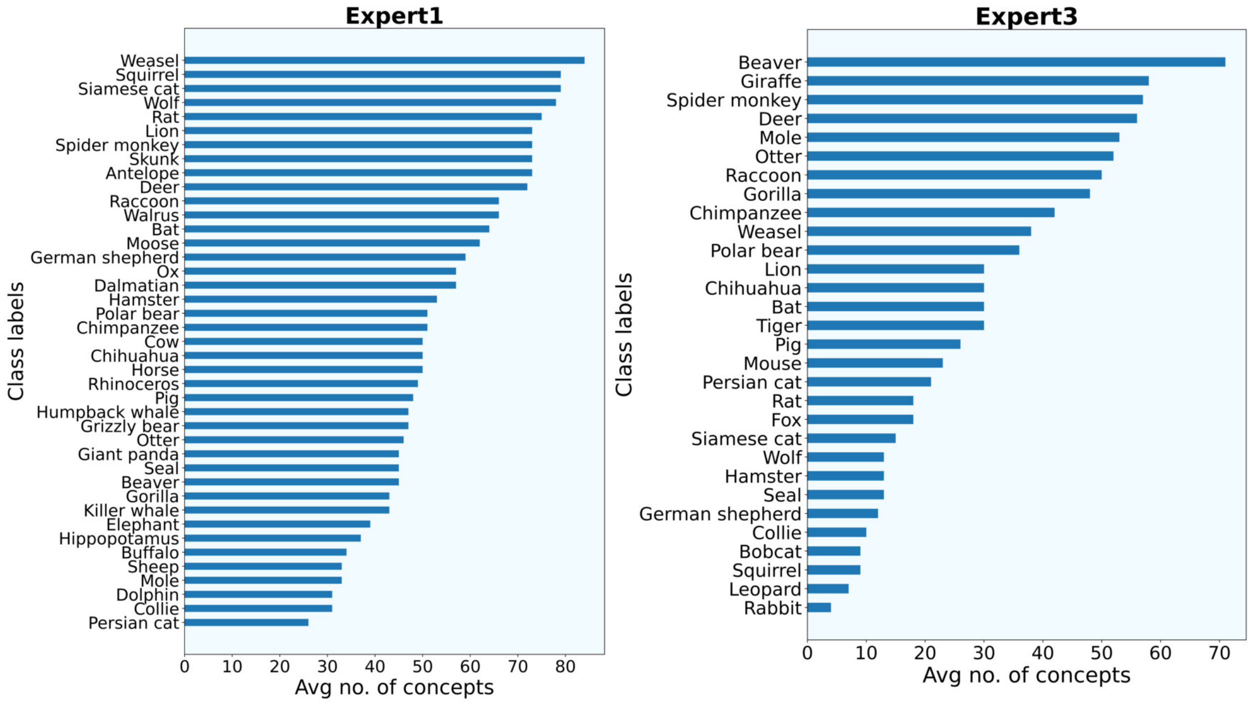
Figure 30. Class labels (Animal species) vs. avg concepts using VIT as the backbone for Awa2. Each bar in this plot indicates the average number of concepts required to explain each sample of that animal species correctly. For example according to (c) expert3 requires approximately 10 concepts to explain an instance of “Antelope”.



(e)

(f)

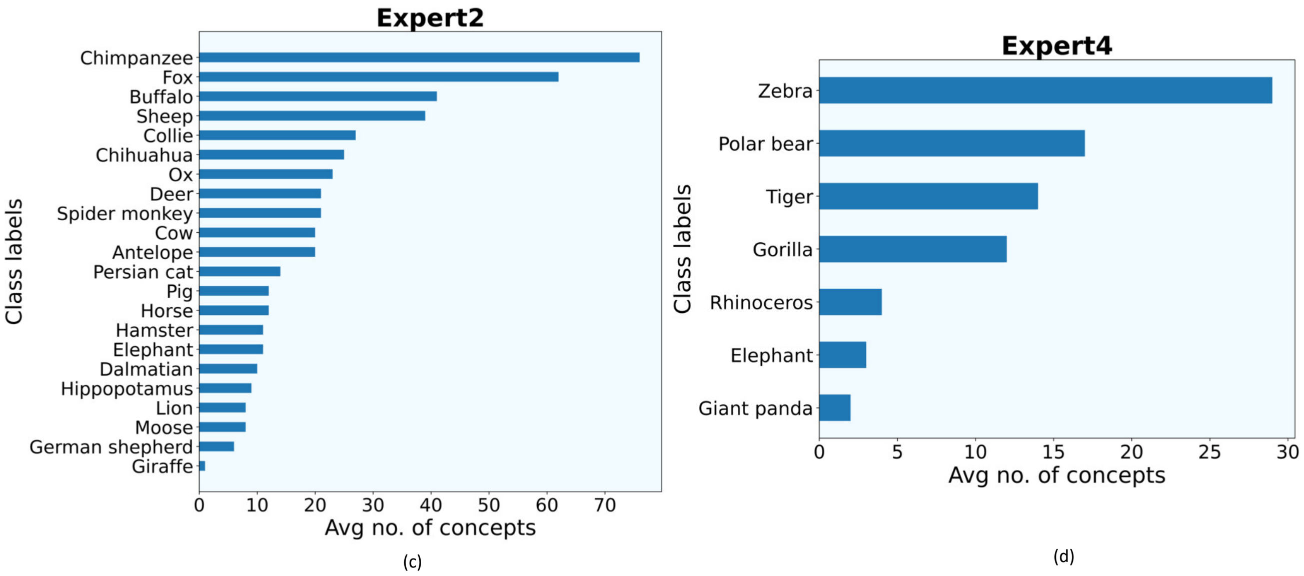
Figure 31. Class labels (Animal species) vs. avg concepts using VIT as the backbone for Awa2. Each bar in this plot indicates the average number of concepts required to explain each sample of that animal species correctly. For example according to (e) expert5 requires approximately 4 concepts to explain an instance of “Antelope”.



(a)

(b)

Figure 32. Class labels (Animal species) vs. avg concepts using ResNet-101 as the backbone for Awa2. Each bar in this plot indicates the average number of concepts required to explain each sample of that animal species correctly. For example according to (a) expert1 requires approximately 80 concepts to explain an instance of “Weasel”.



(c)

(d)

Figure 33. Class labels (Animal species) vs. avg concepts using ResNet-101 as the backbone for Awa2. Each bar in this plot indicates the average number of concepts required to explain each sample of that animal species correctly. For example according to (b) expert2 requires approximately 72 concepts to explain an instance of "Chimpanzee".

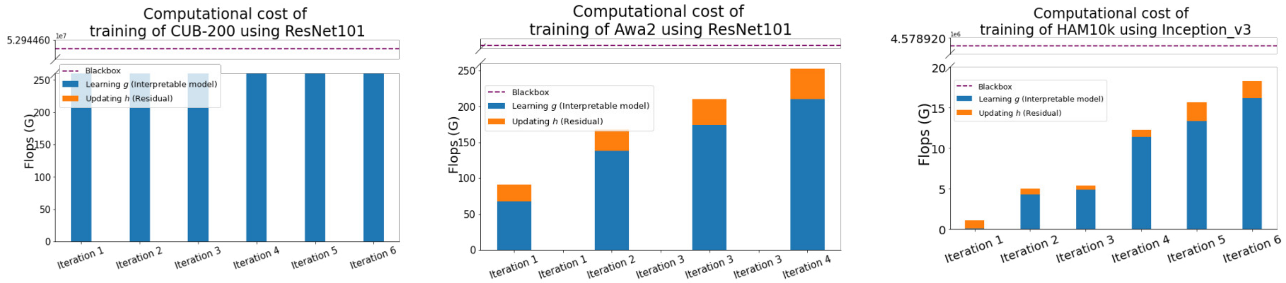


Figure 34. Flops vs. iteration for MoIE and the Blackbox. The dotted line in the figure represents the flops taken by the blackbox.