WHAT DO YOU SEE IN COMMON? LEARNING HIERARCHICAL PROTOTYPES OVER TREE-OF-LIFE TO DISCOVER EVOLUTIONARY TRAITS

Anonymous authors
Paper under double-blind review

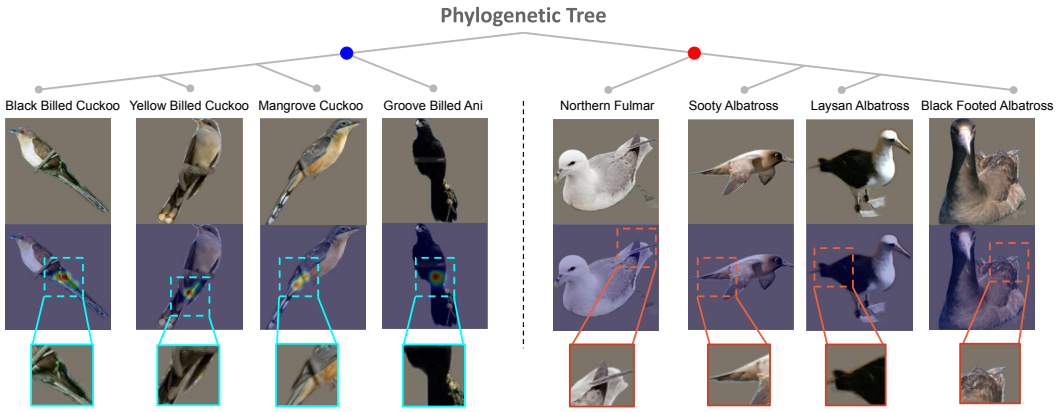


Figure 1: Sample images of bird species with zoomed-in views of learned prototypes along with their associated score maps. We consider the problem of finding evolutionary traits common to a group of species derived from the same ancestor (blue) that are absent in other species from a different ancestor (red). We can infer that descendants of the blue node share a common trait: *long tail*, absent from descendants of the red node.

ABSTRACT

A grand challenge in biology is to discover evolutionary traits—features of organisms common to a group of species with a shared ancestor in the tree of life (also referred to as phylogenetic tree). With the growing availability of image repositories in biology, there is a tremendous opportunity to discover evolutionary traits directly from images in the form of a hierarchy of prototypes. However, current prototype-based methods are mostly designed to operate over a flat structure of classes and face several challenges in discovering hierarchical prototypes, including the issue of learning over-specific prototypes at internal nodes. To overcome these challenges, we introduce the framework of **H**ierarchy aligned **Com**monality through **P**rototypical **Net**works **(HComP-Net)**. The key novelties in HComP-Net include a novel *over-specificity loss* to avoid learning over-specific prototypes, a novel *discriminative loss* to ensure prototypes at an internal node are absent in the contrasting set of species with different ancestry, and a novel *masking module* to allow for the exclusion of over-specific prototypes at higher levels of the tree without hampering classification performance. We empirically show that HComP-Net learns prototypes that are accurate, semantically consistent, and generalizable to unseen species in comparison to baselines.

1 INTRODUCTION

A central goal in biology is to discover the observable characteristics of organisms, or *traits* (e.g., beak color, stripe pattern, and fin curvature), that help in discriminating between species and understanding how organisms evolve and adapt to their environment (Houle & Rossoni, 2022). For example, discovering traits inherited by a group of species that share a common ancestor on the

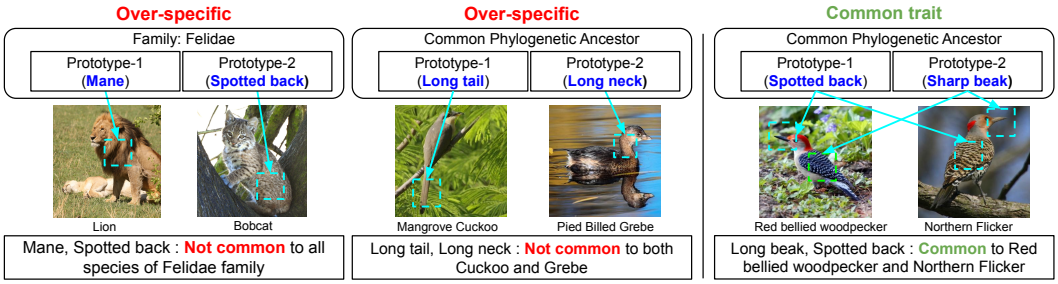


Figure 2: Examples to illustrate the problem of learning "over-specific" prototypes at internal nodes, which only cover one descendant species of the node instead of learning prototypes *common* to all descendants.

tree of life (also referred to as the *phylogenetic tree*, see Figure 1) is of great interest to biologists to understand how organisms diversify and evolve (O'Leary & Kaufman, 2011) (see Appendix A for more details on the biological motivation of this problem). However, the measurement of such traits with evolutionary signals, termed *evolutionary traits*, is not straightforward and often relies on subjective and labor-intensive human expertise and definitions (Simões et al., 2017; Sereno, 2007), hindering rapid scientific advancement (Lürig et al., 2021).

With the growing availability of large-scale image repositories in biology containing tens of thousands of species of organisms (Van Horn et al., 2018; Singer et al., 2018; Wah et al., 2011), there is an opportunity for machine learning (ML) methods to discover evolutionary traits automatically from images, especially those that differentiate visually or genetically similar species (Lürig et al., 2021; Elhamod et al., 2023). This is especially true in light of recent advances in the field of explainable ML, such as the seminal work of ProtoPNet (Chen et al., 2019) and its variants (Rymarczyk et al., 2021; 2022; Nauta et al., 2021) which find representative patches in training images (termed *prototypes*) capturing discriminatory features for every class. We can thus cast the problem of discovering evolutionary traits into asking the following question: *what image features or prototypes are common across a group of species with a shared ancestor in the tree of life that are absent in species with a different shared ancestor?*

For example, in Figure 1, we can see that the four species of birds on the left descending from the blue node show the common feature of having "long tails", unlike any of the descendant species of the red node. Learning such common features at every internal node as a hierarchy of prototypes can help biologists generate novel hypotheses of species diversification (e.g., the splitting of blue and red nodes) and accumulation of evolutionary trait changes.

Despite the success of ProtoPNet (Chen et al., 2019) and its variants including HPnet (Hase et al., 2019) that first introduced the idea of learning hierarchical prototypes at every internal node of the tree, there are three main challenges to be addressed while learning hierarchical prototypes for discovering evolutionary traits. *First*, existing methods that learn multiple prototypes for every class are prone to learning "over-specific" prototypes at internal nodes of a tree, which cover only one (or a few) of its descendant species. Figure 2 shows a few examples to illustrate the concept of over-specific prototypes. Consider the problem of learning prototypes common to descendant species of the Felidae family: Lion and Bobcat. If we learn one prototype focusing on the feature of the mane (specific only to Lion) and another prototype focusing on the feature of spotted back (specific only to Bobcat), then these two prototypes taken together can classify all images from the Felidae family. However, they do not represent *common* features shared between Lion and Bobcat and hence are not useful for discovering evolutionary traits. Such over-specific prototypes should be instead pushed down to be learned at lower levels of the tree (e.g., the species leaf nodes of Lion and Bobcat).

Second, while existing methods such as ProtoPShare (Rymarczyk et al., 2021), ProtoPool (Rymarczyk et al., 2022), and ProtoTree (Nauta et al., 2021) allow prototypes to be shared across classes for re-usability and sparsity, in the problem of discovering evolutionary traits, we want to learn prototypes at an internal node n that are not just shared across all its descendant species but are also absent in the *contrasting set* of species (i.e., species descending from sibling nodes of n representing alternate paths of diversification). *Third*, at higher levels of the tree, finding features that are common across a large number of diverse species is challenging (Harmon et al., 2010; Pennell et al., 2015).

In such cases, we should be able to abstain from finding common prototypes without hampering accuracy at the leaf nodes—a feature missing in existing methods.

To address these challenges, we present **H**ierarchy aligned **Com**monality through **P**rototypical **Net**works **(HComP-Net)**, a framework to learn hierarchical prototypes over the tree of life for discovering evolutionary traits. Here are the main contributions of our work:

1. HComP-Net learns common traits shared by all descendant species of an internal node and avoids the learning of over-specific prototypes in contrast to baseline methods using a novel *over-specificity loss*.
2. HComP-Net uses a novel *discriminative loss* to ensure that the prototypes learned at an internal node are absent in the contrasting set of species with different ancestry.
3. HComP-Net includes a novel *masking module* to allow for the exclusion of over-specific prototypes at higher levels of the tree without hampering classification performance.
4. We empirically show that HComP-Net learns prototypes that are accurate, semantically consistent, and generalizable to unseen species compared to baselines on data from 190 species of birds (CUB-200-2011 dataset) (Wah et al., 2011), 38 species of fishes (Elhamod et al., 2023), 30 species of butterflies (Lawrence & Campolongo, 2024), 113 species of spiders and 41 species of turtles (Van Horn et al., 2021). We show the ability of HComP-Net to generate novel hypotheses about evolutionary traits at different levels of the phylogenetic tree of organisms.

2 Related Works

One of the seminal lines of work in the field of prototype-based interpretability methods is the framework of ProtoPNet (Chen et al., 2019) that learns a set of "prototypical patches" from training images of every class to enable case-based reasoning. Following this work, several variants have been developed, such as ProtoPShare (Rymarczyk et al., 2021), ProtoPool (Rymarczyk et al., 2022), ProtoTree (Nauta et al., 2021), and HPnet (Hase et al., 2019) suiting to different interpretability requirements. Among all these approaches, our work is closely related to HPnet (Hase et al., 2019), the hierarchical extension of ProtoPNet that learns a prototype layer for every parent node in the tree. Despite sharing similar motivation, HPnet is not designed to avoid the learning of over-specific prototypes or to abstain from learning common prototypes at higher levels of the tree.

Another related line of work is the framework of PIPNet (Nauta et al., 2023), which uses self-supervised learning to reduce the "semantic gap" (Hoffmann et al., 2021; Kim et al., 2022) between the latent space of prototypes and the space of images, such that the prototypes in latent space correspond to the same visual concept in the image space. In HComP-Net, we build upon the idea of self-supervised learning introduced in PIPNet to learn a semantically consistent hierarchy of prototypes. Our work is also related to ProtoTree (Nauta et al., 2021), which structures the prototypes as nodes in a decision tree to offer more granular interpretability. However, ProtoTree differs from our work in that it learns the tree-based structure of prototypes automatically from data and cannot handle a known hierarchy. Moreover, the prototypes learned in ProtoTree are purely discriminative and allow for negative reasoning, which is not aligned with our objective of finding common traits of descendant species.

Other related works that focus on finding shared features are ProtoPShare (Rymarczyk et al., 2021) and ProtoPool (Rymarczyk et al., 2022). Both approaches aim to find common features among classes, but their primary goal is to reduce the prototype count by exploiting similarities among classes, leading to a sparser network. This is different from our goal of finding a hierarchy of prototypes to find evolutionary traits common to a group of species absent in other species.

Outside the realm of prototype-based methods, the framework of Phylogeny-guided Neural Networks (PhyloNN) (Elhamod et al., 2023) shares a similar motivation as our work to discover evolutionary traits by representing biological images in feature spaces structured by tree-based knowledge (i.e., phylogeny). However, PhyloNN primarily focuses on the tasks of image generation and translation rather than interpretability. Additionally, PhyloNN can only work with discretized trees with fixed number of ancestor levels per leaf node, unlike our work that does not require any discretization of the tree.

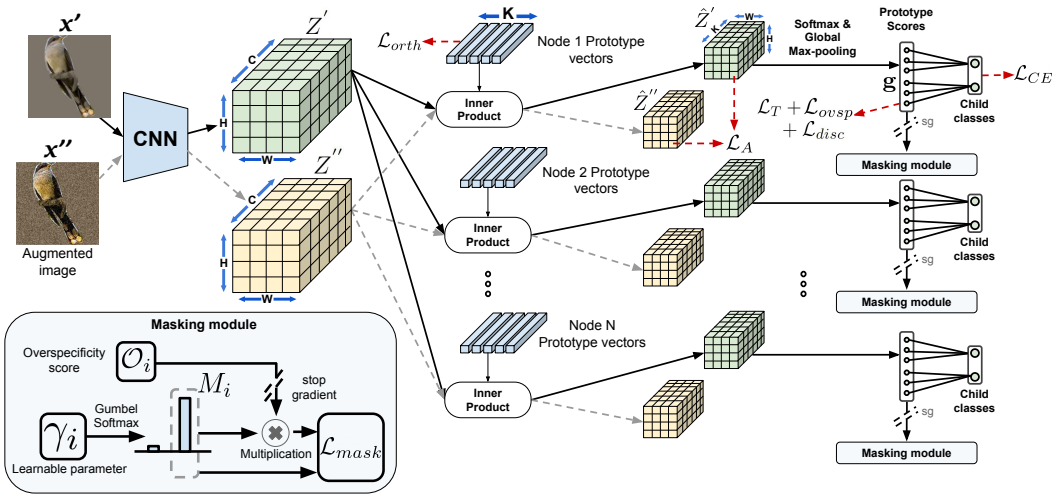


Figure 3: Schematic illustration of HComP-Net model architecture.

There are also methods for learning hierarchical features in image classification that can classify difficult-to-distinguish categories within coarse-grained datasets (Yan et al., 2015; Hu et al., 2016; Taherkhani et al., 2019). These methods use semantic hierarchies that separate broad super-categories and sub-categories instead of enforcing a multi-level hierarchy among fine-grained images (for example, images of birds only). Other related works include regularization techniques to capture commonalities between similar classes Goo et al. (2016), and hierarchical interpretability frameworks Dai et al. (2023) that extend the concept of concept whitening (Chen et al., 2020) to incorporate known hierarchies. However, the interpretability offered by these approaches cannot localize to fine-grained image parts. In contrast, our approach enforces a multi-level predefined hierarchy on images, and learns prototypes that represent localized attributes common to different nodes of the hierarchy.

3 Proposed Methodology

3.1 HComP-Net Model Architecture

Given a phylogenetic tree with N internal nodes, the goal of HComP-Net is to jointly learn a set of prototype vectors $\mathbf{P_n}$ for every internal node $n \in \{1, \ldots, N\}$. Our architecture as shown in Figure 3 begins with a CNN that acts as a common feature extractor $f(x; \theta)$ for all nodes, where θ represents the learnable parameters of f. f converts an image x into a latent representation $Z \in \mathbb{R}^{H \times W \times C}$, where each "patch" at location (h, w) is, $\mathbf{z}_{\text{h,w}} \in \mathbb{R}^C$. Following the feature extractor, for every node n, we initialize a set of K_n prototype vectors $\mathbf{P_n} = \{\mathbf{p_i}\}_{i=1}^{K_n}$, where $\mathbf{p_i} \in \mathbb{R}^C$. Here, the number of prototypes K_n learned at node n varies in proportion to the number of children of node n, with β as the proportionality constant, i.e., at each node n we assign β prototypes for every child node. To simplify notations, we drop the subscript n in $\mathbf{P_n}$ and K_n while discussing the operations occurring in node n.

We consider the following sequence of operations at every node n. We first compute the similarity score between every prototype in $\mathbf{P}$ and every patch in Z. This results in a matrix $\hat{Z} \in \mathbb{R}^{H \times W \times K}$, where every element represents a similarity score between image patches and prototype vectors. We apply a softmax operation across the K channels of $\hat{Z}$ such that the vector $\hat{\mathbf{z}}_{\text{h,w}} \in \mathbb{R}^K$ at spatial location (h, w) in $\hat{Z}$ represents the probability that the corresponding patch $\mathbf{z}_{\text{h,w}}$ is similar to the K prototypes. Furthermore, the ith channel of $\hat{Z}$ serves as a prototype score map for the prototype vector $\mathbf{p_i}$, indicating the presence of $\mathbf{p_i}$ in the image. We perform global max-pooling across the spatial dimensions $H \times W$ of $\hat{Z}$ to obtain a vector $\mathbf{g} \in \mathbb{R}^K$, where the ith element represents the highest similarity score of the prototype vector $\mathbf{p_i}$ across the entire image. $\mathbf{g}$ is then fed to a linear classification layer with weights ϕ to produce the final classification scores for every child

node of node n. We restrict the connections in the classification layer so that every child node n_c is connected to a distinct set of β prototypes, to ensure that every prototype uniquely maps to a child node. ϕ is restricted to be non-negative to ensure that the classification is done solely through positive reasoning, similar to the approach used in PIP-Net (Nauta et al., 2023). We borrow the regularization scheme of PIP-Net to induce sparsity in ϕ by computing the logit of child node n_c as $\log((\mathbf{g}\phi)^2 + 1)$. $\mathbf{g}$ and ϕ here are again unique to each node.

3.2 Loss Functions Used to Train HComP-Net

Contrastive Losses for Learning Hierarchical Prototypes: PIP-Net (Nauta et al., 2023) introduced the idea of using self-supervised contrastive learning to learn semantically meaningful prototypes. We build upon this idea in our work to learn semantically meaningful hierarchical prototypes at every node in the tree as follows. For every input image $\mathbf{x}$, we pass in two augmentations of the image, $\mathbf{x}'$ and $\mathbf{x}''$ to our framework. The prototype score maps for the two augmentations, $\hat{Z}^{'}$ and $\hat{Z}^{''}$, are then considered as positive pairs. Since $\hat{\mathbf{z}}_{\text{h,w}} \in \mathbb{R}^K$ represents the probabilities of patch $\mathbf{z}_{\text{h,w}}$ being similar to the prototypes from $\mathbf{P}$, we align the probabilities from the two augmentations $\hat{\mathbf{z}}^{'}_{\text{h,w}}$ and $\hat{\mathbf{z}}^{''}_{\text{h,w}}$ to be similar using the following alignment loss:

$$\mathcal{L}_A = -\frac{1}{HW} \sum_{(h,w) \in H \times W} \log(\hat{\mathbf{z}}^{'}_{\mathbf{h,w}} \cdot \hat{\mathbf{z}}^{''}_{\mathbf{h,w}}) \tag{1}$$

Since $\sum_{i=1}^{K} \hat{\mathbf{z}}_{\mathbf{h,w,i}} = 1$ due to softmax operation, $\mathcal{L}_A$ is minimum (i.e., $\mathcal{L}_A = 0$) when both $\hat{\mathbf{z}}^{'}_{\mathbf{h,w}}$ and $\hat{\mathbf{z}}^{''}_{\mathbf{h,w}}$ are identical one-hot encoded vectors. A trivial solution that minimizes $\mathcal{L}_A$ is when all patches across all images are similar to the same prototype. To avoid such representation collapse, we use the following tanh-loss $\mathcal{L}_T$ of PIP-Net (Nauta et al., 2023), which serves the same purpose as uniformity losses in Wang & Isola (2020) and Silva & Rivera (2022):

$$\mathcal{L}_T = -\frac{1}{K} \sum_{i=1}^{K} \log(\tanh(\sum_{b=1}^{B} \mathbf{g}_{\mathbf{b,i}})), \tag{2}$$

where $\mathbf{g}_{\mathbf{b,i}}$ is the prototype score for prototype i with respect to image b of mini-batch. $\mathcal{L}_T$ encourages each prototype $\mathbf{p_i}$ to be activated at least once in a given mini-batch of B images, thereby helping to avoid the possibility of representation collapse. The use of $\tanh$ ensures that only the presence of a prototype is taken into account and not its frequency.

Over-specificity Loss: To achieve the goal of learning prototypes common to all descendant species of an internal node, we introduce a novel loss, termed *over-specificity loss* $\mathcal{L}_{ovsp}$ that avoids learning over-specific prototypes at any node n. $\mathcal{L}_{ovsp}$ is formulated as a modification of the tanh-loss such that prototype $\mathbf{p_i}$ is encouraged to be activated at least once in every one of the descendant species $d \in \{1, \ldots, D_i\}$ of its corresponding child node in the mini-batch of images fed to the model, as follows:

$$\mathcal{L}_{ovsp} = -\frac{1}{K} \sum_{i=1}^{K} \sum_{d=1}^{D_i} \log(\tanh(\sum_{b \in B_d} \mathbf{g}_{\mathbf{b,i}})), \tag{3}$$

where B_d is the subset of images in the mini-batch that belong to species d.

Discriminative loss: In order to ensure that a learned prototype for a child node n_c is not activated by any of its *contrasting set* of species (i.e., species that are descendants of child nodes of n other than n_c), we introduce another novel loss function, $\mathcal{L}_{disc}$, defined as follows:

$$\mathcal{L}_{disc} = \frac{1}{K} \sum_{i=1}^{K} \sum_{d \in \widetilde{D_i}} \max_{b \in B_d}(\mathbf{g}_{\mathbf{b,i}}), \tag{4}$$

where $\widetilde{D_i}$ is the contrasting set of all descendant species of child nodes of n other than n_c. This is similar to the separation loss used in other prototype-based methods such as ProtoPNet (Chen et al., 2019), ProtoTree (Nauta et al., 2021), and TesNet (Wang et al., 2021).

Orthogonality loss: We also apply kernel orthogonality as introduced in Wang et al. (2020) to the prototype vectors at every node n, so that the learned prototypes are orthogonal and capture diverse

features:

$$\mathcal{L}_{orth} = \|\hat{\mathbf{P}}\hat{\mathbf{P}}^\top - I\|_F^2 \tag{5}$$

where $\hat{\mathbf{P}}$ is the matrix of normalized prototype vectors of size $C \times K$, I is an identity matrix, and $\|.\|_F^2$ is the Frobenius norm. Each prototype $\hat{\mathbf{p}}_\mathbf{i}$ in $\hat{\mathbf{P}}$ is normalized as, $\hat{\mathbf{p}}_\mathbf{i} = \frac{\mathbf{p_i}}{\|\mathbf{p_i}\|}$.

Classification loss: Finally, we apply cross-entropy loss for classification at each internal node as follows:

$$\mathcal{L}_{CE} = -\sum_{b}^{B} y_b \log(\hat{y}_b) \tag{6}$$

where y is ground truth label and $\hat{y}$ is the prediction at every node of the tree.

3.3 Masking Module to Identify Over-specific Prototypes

We employ an additional masking module at every node n to identify over-specific prototypes without hampering their training. The learned mask for prototype $\mathbf{p_i}$ simply serves as an indicator of whether $\mathbf{p_i}$ is over-specific or not, enabling our approach to abstain from finding common prototypes if there are none, especially at higher levels of the tree. To obtain the mask values, we first calculate the over-specificity score for prototype $\mathbf{p_i}$ as the product of the maximum prototype scores obtained across all images in the mini-batch belonging to every descendant species d as:

$$\mathcal{O}_i = -\prod_{d=1}^{D_i} \max_{(b \in B_d)} (\mathbf{g_{b,i}}) \tag{7}$$

where $\mathbf{g_{b,i}}$ is the prototype score for prototype $\mathbf{p_i}$ with respect to image b of mini-batch and B_d is the subset of images in the mini-batch that belong to descendant species d. Since $\mathbf{g_{b,i}}$ takes a value between 0 to 1 due to the softmax operation, $\mathcal{O}_i$ ranges from -1 to 0, where -1 denotes least over-specificity and 0 denotes the most over-specificity. The multiplication of the prototype scores ensures that even when the score is less with respect to only one descendant species, the prototype will be assigned a high over-specificity score (close to 0).

As shown in Figure 3, $\mathcal{O}_i$ is then fed into the masking module, which includes a learned mask value M_i for every prototype $\mathbf{p_i}$. We generate M_i from a Gumbel-softmax distribution (Jang et al., 2016) so that the values are skewed to be very close to either 0 or 1, i.e., $M_i = \text{Gumbel-Softmax}(\gamma_i, \tau)$, where γ_i are the learnable parameters of the distribution and τ is temperature. We then compute the masking loss, $\mathcal{L}_{mask}$, as:

$$\mathcal{L}_{mask} = \sum_{i=1}^{K} (\lambda_{mask} M_i \circ \texttt{stopgrad}(\mathcal{O}_i) + \lambda_{L_1} \|M_i\|_1) \tag{8}$$

where λ_{mask} and λ_{L_1} are trade-off coefficients, $\|.\|_1$ is the L_1 norm added to induce sparsity in the masks, and `stopgrad` represents the stop gradient operation applied over $\mathcal{O}_i$ to ensure that the gradient of $\mathcal{L}_{mask}$ does not flow back to the learning of prototype vectors and impact their training. Note that *the learned masks are not used for pruning the prototypes during training*, they are only used during inference to determine which of the learned prototypes are over-specific and likely to not represent evolutionary traits. Therefore, even if all the prototypes are identified as over-specific by the masking module at an internal node, it will not affect the classification performance at that node.

3.4 Training HComP-Net

We first pre-train the prototypes at every internal node in a self-supervised learning manner using alignment and tanh-losses as $\mathcal{L}_{SS} = \lambda_A \mathcal{L}_A + \lambda_T \mathcal{L}_T$. We then fine-tune the model using the following combined loss: $(\lambda_{CE}\mathcal{L}_{CE} + \mathcal{L}_{SS} + \lambda_{ovsp}\mathcal{L}_{ovsp} + \lambda_{disc}\mathcal{L}_{disc} + \lambda_{orth}\mathcal{L}_{orth} + \mathcal{L}_{mask})$, where λ's are trade-off parameters. Note that the loss is applied over every node in the tree. We show an ablation of key loss terms used in our framework in Table 6 in Appendix D.

4 Experimental Setup

Datasets: In our experiments, we primarily focus on the 190 species of birds (**Bird**) from the CUB-200-2011 (Wah et al., 2011) dataset for which the phylogenetic relationship (Jetz et al., 2012) is

known. The tree is quite large with a total of 184 internal nodes. We removed the background from the images to avoid the possibility of learning prototypes corresponding to background information, such as the bird's habitat, as we are only interested in the traits corresponding to the body of the organism. We also apply our method on a fish dataset with 38 species (**Fish**) (Elhamod et al., 2023) along with its associated phylogeny (Elhamod et al., 2023), 30 subspecies of Heliconius butterflies (**Butterfly**) from the Jiggins Heliconius Collection dataset (Lawrence & Campolongo, 2024) collected from various sources (see Appendix F) . We also consider certain subsets of iNat2021 dataset (Van Horn et al., 2021) where the set of species are not too diverse (such as elephants and whales from class Mammalia), but exhibit fine-grained, hard-to-quantify hierarchical variation in visually observable traits, such that we can find interpretable commonalities between the species. We reiterate that our goal is to discover *local* prototypes as evolutionary traits, e.g., beak color, stripe pattern, and fin curvature. Traits differentiating species at higher levels of the tree are often not local, such as global contours, which are beyond the scope of our study. Therefore, we consider 41 species from order *Testudines* (**Turtle**) and 113 species from order *Araneae* (**Spider**) in the iNat2021 dataset. The phylogeny for Butterfly, Turtle, and Spider were obtained using the `rotl` package (Redelings et al., 2019; Michonneau et al., 2016). Further details of dataset statistics, phylogeny statistics, hyper-parameter settings and training strategy are provided in Appendix F.

Baselines: We compare HComP-Net to ResNet-50 (He et al., 2016), ConvNeXt-tiny(Liu et al., 2022), ProtoPNet (Chen et al., 2019), INTR (Interpretable Transformer) (Paul et al., 2023), PIP-Net (Nauta et al., 2023) and HPnet (Hase et al., 2019) as classification baselines. For HPnet, we used the same hyperparameter settings and training strategy as used by ProtoPNet for the CUB-200-2011 dataset. To ensure a fair comparison, we set the number of prototypes per child class to 10 for both HComP-Net and HPnet on the Bird, Butterfly, and Fish datasets. For the Spider and Lizard datasets, due to the higher diversity of images, we increased the number of prototypes to 20 for both methods. We follow the same training strategy as provided by ProtoPNet for the CUB-200-2011 dataset.

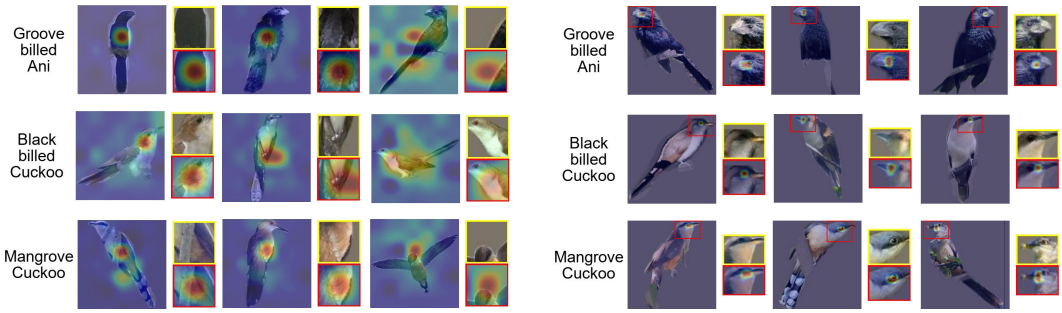
5 Results

5.1 Fine-grained Accuracy

Similar to HPnet (Hase et al., 2019), we calculate the fine-grained accuracy for each leaf node by calculating the path probability over every image. During inference, the final probability for leaf class Y given an image X is calculated as, $P(Y|X) = P(Y^{(1)}, Y^{(2)}, ..., Y^{(L)}|X) = \prod_{l=1}^{L} P(Y^{(l)}|X)$, where $P(Y^{(l)}|X)$ is the probability of assigning image X to a node at level l, and L is the depth of the leaf node. Every image is assigned to the leaf class with maximum path probability, which is used to compute the fine-grained accuracy. The comparison of the fine-grained accuracy calculated for HComP-Net and the baselines are given in Table 1 (best performance bolded and second best performance underlined). Note that despite extensive hyper-parameter tuning, HPnet was unable to perform well on Spider and Turtle datasets (See Appendix F for details). It is important to note that the *primary objective of our work is identifying evolutionary traits through semantically meaningful prototypes* (both quantitatively and qualitatively evaluated in Section 5.3 and 5.4 respectively), and not necessarily achieving high classification accuracy (which may require capturing global features or features prone to dataset bias). Additionally, we observe that the Spider and Turtle datasets from iNat2021 are relatively more challenging since they contain noisy elements, including blurry images, obscured foregrounds and camouflaged backgrounds, that cause models to misclassify these images (examples provided in Figure 7 of Appendix).

Table 1: % Fine-grained Accuracy

Model	Hierarchy	Bird	Butterfly	Fish	Spider	Turtle
ResNet-50	No	74.18	95.76	86.63	74.17	56.23
ConvNeXt-tiny		**84.23**	96.28	<u>91.73</u>	**84.05**	**69.56**
ProtoPNet		75.80	96.82	91.11	74.17	61.74
INTR		69.22	95.53	86.73	78.91	58.64
PIPNet		<u>83.38</u>	**97.87**	**93.58**	<u>83.45</u>	<u>67.24</u>
HPnet	Yes	36.18	94.69	77.51	5.85	6.38
HComP-Net		70.01	<u>97.35</u>	90.80	76.19	58.26



(a) HPnet (b) HComP-Net

Figure 4: Comparing the part consistency of HPnet and HComP-Net for their prototype learned at an internal node in the **Bird** dataset that corresponds to 3 descendant species (names shown on the rows). For every species, we are visualizing the top-3 images with highest prototype score for both HPnet and HComP-Net, shown as the three columns with zoomed in views of their discovered prototypes. We can see that *HPnet highlights varying parts of the bird* across the 3 species and across multiple images of the same species, making it difficult to associate a consistent semantic meaning to its learned prototype. In contrast, *HComP-Net consistently highlights the head region* of the bird across all four species and their images.

5.2 Generalizing to Unseen Species in the Phylogeny

We analyze the performance of HComP-Net in generalizing to unseen species that the model has not seen during training. The biological motivation for this experiment is to evaluate if HComP-Net can situate newly discovered species at its appropriate position in the phylogeny by identifying its common ancestors shared with the known species. An added advantage of our work is that along with identifying the ancestor of an unseen species, we can also identify the common traits shared by the novel species with known species in the phylogeny.

Table 2: % Fine-grained Accuracy (on unseen species)

Species Name	HComP-Net	HPnet
Fish Crow	53.33	10.55
Rock Wren	53.33	10.22
Indigo Bunting	96.67	49.2
Bohemian Waxwing	70.00	44.9

Since unseen species cannot be classified to the finest levels (i.e., up to the leaf node corresponding to the unseen species), we analyze the ability of HComP-Net to classify unseen species accurately up to one level above the leaf level in the hierarchy. With this consideration, the final probability of an unseen species for a given image is calculated as, $P(Y|X_{unseen}) = P(Y^{(1)}, Y^{(2)}, ..., Y^{(L-1)}|X) = \prod_{l=1}^{L-1} P(Y^{(l)}|X)$. Note that we leave out the class probability at the Lth level, since we do not take into account the class probability of the leaf level. Since such path probability can only be calculated for hierarchical methods, we compare our approach to HPnet which also learns hierarchical prototypes. We leave four species from the Bird training set and calculate their accuracy during inference in Table 2. We can see that HComP-Net is able to generalize better than HPnet for all four species.

5.3 Analyzing the Semantic Quality of Prototypes

Following the method introduced in PIPNet (Nauta et al., 2023), we assess the semantic quality of our learned prototypes by evaluating their part purity. A prototype with high part purity (close to 1) is one that consistently highlights the same image region in the score maps (corresponding to consistent local features such as the eye or wing of a bird) across images belonging to the same class.

Table 3: Part purity of prototypes on **Bird** dataset.

Model	$\mathcal{L}_{ovsp}$	Masking	Part purity	% masked
HPnet	-	-	0.14 ± 0.09	-
HComP-Net	-	-	0.68 ± 0.22	-
HComP-Net	-	✓	0.75 ± 0.17	21.42%
HComP-Net	✓	-	0.72 ± 0.19	-
HComP-Net	✓	✓	$\mathbf{0.77 \pm 0.16}$	16.53%

The part purity is calculated using the part locations of 15 parts that are provided in the CUB dataset. For each prototype, we take the top-10 images from each leaf descendant. We consider the 32×32

image patch that is centered around the max activation location of the prototype from the top-10 images. With these top-10 image patches, we calculate how frequently each part is present inside the image patch. For example, a part that is found inside the image patch 8 out of 10 times is given a score of 0.8. In PIP-Net, the highest value among the values calculated for each part is given as the part purity of the prototype. In our approach, since we are dealing with a hierarchy and taking the top-10 from each leaf descendant, a particular part, let's say the eye, might have a score of 0.5 for one leaf descendant and 0.7 for a different leaf descendant. Since we want the prototype to represent the same part for all the leaf descendants, we take the lowest score (the weakest link) among all the leaf descendants as the score of the part. By following this method, for a given prototype we can arrive at a value for each part and finally take the maximum among the values as the purity of the prototype. We take the mean of the part purity across all the prototypes and report the results in Table 3 for different ablations of HComP-Net and also HPnet, which is the only baseline method that can learn hierarchical prototypes.

We can see that HComP-Net, even without the use of over-specificity loss, performs much better than HPnet due to the contrastive learning approach we have adopted from PIPNet (Nauta et al., 2023). The addition of over-specificity loss improves the part purity because over-specific prototypes tend to have poor part purity for some of the leaf descendants, which will affect their overall part purity score. Further, for both ablations with and without over-specificity loss, we apply the masking module and remove masked (over-specific) prototypes during the calculation of part purity. We see that the part purity goes higher by applying the masking module, demonstrating its effectiveness in identifying over-specific prototypes. We further compute the purity of masked-out prototypes and notice that the masked-out prototypes have drastically lower part purity (0.29 ± 0.17) compared to non-masked prototypes (0.77 ± 0.16). We also provide a visual comparison of a masked (over-specific) prototype and an unmasked (non-over-specific) prototype in Appendix H. An alternative approach to learning the masking module is to identify over-specific prototypes using a fixed global threshold over $\mathcal{O}_i$. We show in Table 9 of Appendix G that given the right choice of such a threshold, we can identify over-specific prototypes. However, selecting the ideal threshold can be non-trivial. On the other hand, our masking module learns the appropriate threshold dynamically as part of the training process.

Figure 4 visualizes the part consistency of prototypes discovered by HComP-Net in comparison to HPnet for the bird dataset. We can see that HComP-Net is finding a consistent region in the image (corresponding to the head region) across all three descendant species and all images of a species, in contrast to HPnet. Since HPnet is implemented with a 7×7 latent space in comparison to HComP-Net which uses a 26×26 latent space, we explore HPnet with a higher resolution feature map of 28×28 and report the qualitative and quantitative results in Figure 9 in Appendix I. We note that the performance of HPnet does not improve with higher resolution feature maps. Furthermore, thanks to the alignment loss, in HComP-Net every patch $\hat{\mathbf{z}}_{\text{h,w}}$ is encoded as nearly a one-hot encoding with respect to the K prototypes which causes the prototype score maps to be highly localized. The concise and focused nature of the prototype score maps makes the interpretation much more effective compared to baselines.

5.4 Analyzing Evolutionary Traits Discovered by HComP-Net

We now qualitatively analyze some of the hypothesized evolutionary traits discovered in the hierarchy of prototypes learned by HComP-Net. Figure 5 shows the hierarchy of prototypes discovered over a small subtree of the phylogeny from Bird (four species) and Fish (three species) dataset. In the visualization of bird prototypes, we can see that the two Pelican species share a consistent region in the learned Prototype labeled 2, which corresponds to the head region of the birds. We can hypothesize this prototype is capturing the white-colored crown common to the two species. On the other hand, Prototype 1 finds the shared trait of similar beak morphology (e.g., sharpness of beaks) across the two Cormorant species. We can see that HComP-Net avoids the learning of over-specific prototypes at internal nodes, which are pushed down to individual leaf nodes, as shown in visualizations of Prototype 3, 4, 5, and 6. Similarly, in the visualization of the fish prototypes, we can see that Prototype 1 is highlighting a specific fin (dorsal fin) of the *Carassius auratus* and *Notropis hudsonius* species, possibly representing their pigmentation and structure, which is noticeably different compared to the contrasting species of *Alosa chrysochloris*. Note that while HComP-Net identifies the common regions corresponding to each prototype (shown as heatmaps), the textual descriptions of the traits provided in Figure 5 are based on human interpretation.

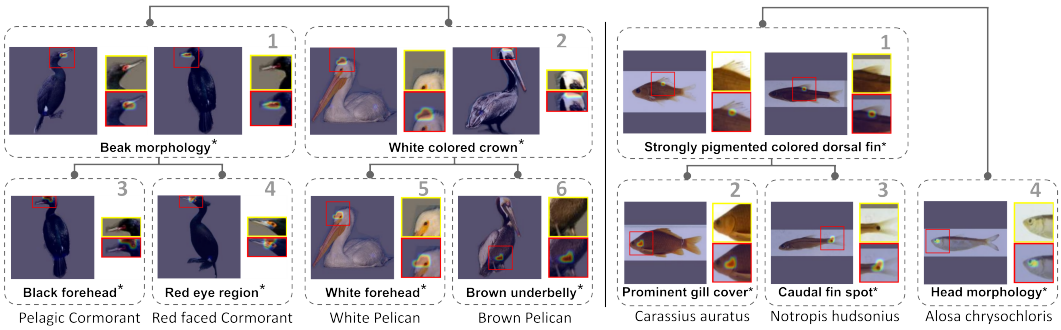


Figure 5: Visualizing the hierarchy of prototypes discovered by HComP-Net for **Bird**s and **Fish**es. *Note that the textual descriptions of the hypothesized traits shown for every prototype are based on human interpretation. Visualizations for **Butterfly**, **Turtle**, and **Spider** datasets are provided in the Appendix J.

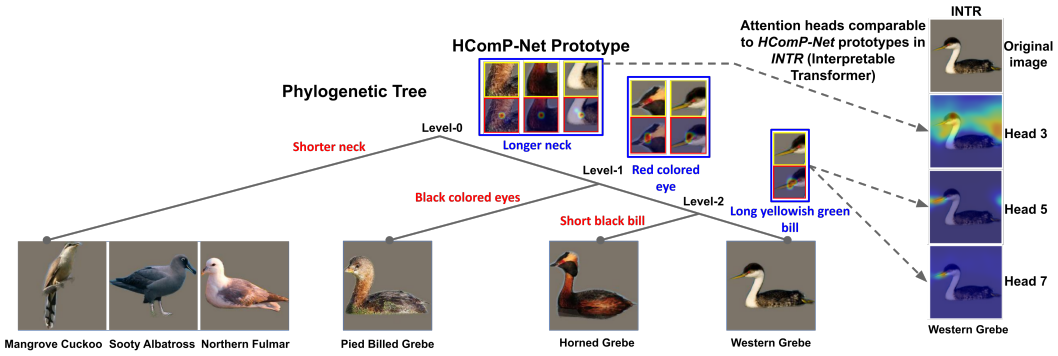


Figure 6: We trace the prototypes learned for Western Grebe at three different levels in the phylogenetic tree (corresponding to different periods of time in evolution). Text in blue is the interpretation of common traits of descendants found by HComP-Net at every ancestor node of Western Grebe.

Figure 6 shows another visualization of the sequence of prototypes learned by HComP-Net for the Western Grebe species at different levels of the phylogeny. We can see that at level 0, we are capturing features closer to the neck region, indicating the likely difference between the length of necks between Grebe species and other species (Cuckoo, Albatross, and Fulmar) that diversify at an earlier time in the process of evolution. At level 1, the prototype is focusing on the eye region, potentially indicating a difference in the color of red and black patterns around the eyes. At level 2, we are differentiating Western Grebe from Horned Grebe based on the feature of bills. We also validate our prototypes by comparing them with the multi-head cross-attention maps learned by INTR (Paul et al., 2023). We can see that some of the prototypes discovered by HComP-Net can be mapped to equivalent attention heads of INTR. However, while INTR is designed to produce a flat structure of attention maps, we are able to place these maps on the tree of life. This shows the power of HComP-Net in generating novel hypotheses about how trait changes may have evolved and accumulated across different branches of the phylogeny. Additional visualizations of discovered evolutionary traits from all five datasets are provided in the Appendix (Figures 10 to 29).

6 Conclusion

We introduce a novel approach for learning hierarchy-aligned prototypes while avoiding the learning of over-specific features at internal nodes of the phylogenetic tree, enabling the discovery of novel evolutionary traits. Our empirical analysis on birds, fishes, and butterflies demonstrates the efficacy of HComP-Net over baseline methods. Furthermore, HComP-Net demonstrates a unique ability to generate novel hypotheses about evolutionary traits, showcasing its potential in advancing our understanding of evolution. We discuss the limitations of our work in Appendix M. While we focus on the biological problem of discovering evolutionary traits, our work can be applied in general to domains involving a hierarchy of classes, which can be explored in future research.

REPRODUCIBILITY

We have ensured the reproducibility of our work by providing a detailed set of resources. The source code used for training and evaluation, along with instructions for reproducing the experiments, is available via an anonymous GitHub link (https://anonymous.4open.science/r/HComP-Net-ICLR-14F6/), which is mentioned in Appendix L. Comprehensive implementation details, including hyperparameters and compute resources, are described in Appendix F. Additionally, we have cited all dataset sources and outlined the data processing steps in Appendix F to facilitate accurate replication of our experiments.

REFERENCES

Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019.

Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020.

Haixing Dai, Lu Zhang, Lin Zhao, Zihao Wu, Zhengliang Liu, David Liu, Xiaowei Yu, Yanjun Lyu, Changying Li, Ninghao Liu, et al. Hierarchical semantic tree concept whitening for interpretable image classification. *arXiv preprint arXiv:2307.04343*, 2023.

Mohannad Elhamod, Mridul Khurana, Harish Babu Manogaran, Josef C Uyeda, Meghan A Balk, Wasila Dahdul, Yasin Bakis, Henry L Bart Jr, Paula M Mabee, Hilmar Lapp, et al. Discovering novel biological traits from images using phylogeny-guided neural networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3966–3978, 2023.

R. Farrell. Cub-200-2011 segmentations (1.0) [data set], 2024. URL `https://data.caltech.edu/records/w9d68-gec53`.

Wonjoon Goo, Juyong Kim, Gunhee Kim, and Sung Ju Hwang. Taxonomy-regularized semantic deep convolutional neural networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pp. 86–101. Springer, 2016.

Luke J Harmon, Jonathan B Losos, T Jonathan Davies, Rosemary G Gillespie, John L Gittleman, W Bryan Jennings, Kenneth H Kozak, Mark A McPeek, Franck Moreno-Roark, Thomas J Near, et al. Early bursts of body size and shape evolution are rare in comparative data. *Evolution*, 64 (8):2385–2396, 2010.

Peter Hase, Chaofan Chen, Oscar Li, and Cynthia Rudin. Interpretable image recognition with hierarchical prototypes. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 7, pp. 32–40, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Adrian Hoffmann, Claudio Fanconi, Rahul Rade, and Jonas Kohler. This looks like that... does it? shortcomings of latent space prototype interpretability in deep networks. *arXiv preprint arXiv:2105.02968*, 2021.

David Houle and Daniela M Rossoni. Complexity, evolvability, and the process of adaptation. *Annual Review of Ecology, Evolution, and Systematics*, 53, 2022.

Hexiang Hu, Guang-Tong Zhou, Zhiwei Deng, Zicheng Liao, and Greg Mori. Learning structured inference neural networks with label relations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2960–2968, 2016.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

W. Jetz, G. H. Thomas, J. B. Joy, K. Hartmann, and A. O. Mooers. The global diversity of birds in space and time. *Nature*, 491:444–448, 2012. URL http://birdtree.org.

Chris Jiggins and Ian Warren. Cambridge butterfly wing collection - Chris Jiggins 2001/2 broods batch 1, January 2019a. URL https://doi.org/10.5281/zenodo.2549524.

Chris Jiggins and Ian Warren. Cambridge butterfly wing collection - Chris Jiggins 2001/2 broods batch 2, January 2019b. URL https://doi.org/10.5281/zenodo.2550097.

Chris Jiggins, Gabriela Montejo-Kovacevich, Ian Warren, and Eva Wiltshire. Cambridge butterfly wing collection batch 3, May 2019. URL https://doi.org/10.5281/zenodo.2682458.

Anuj Karpatne, Xiaowei Jia, and Vipin Kumar. Knowledge-guided machine learning: Current trends and future prospects. *arXiv preprint arXiv:2403.15989*, 2024.

Mridul Khurana, Arka Daw, M Maruf, Josef C Uyeda, Wasila Dahdul, Caleb Charpentier, Yasin Bakış, Henry L Bart Jr, Paula M Mabee, Hilmar Lapp, et al. Hierarchical conditioning of diffusion models using tree-of-life for studying species evolution. *arXiv preprint arXiv:2408.00160*, 2024.

Sunnie SY Kim, Nicole Meister, Vikram V Ramaswamy, Ruth Fong, and Olga Russakovsky. Hive: Evaluating the human interpretability of visual explanations. In *European Conference on Computer Vision*, pp. 280–298. Springer, 2022.

Christopher Lawrence and Elizabeth G. Campolongo. Heliconius collection (cambridge butterfly), 2024. URL https://huggingface.co/datasets/imageomics/Heliconius-Collection_Cambridge-Butterfly.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11976–11986, 2022.

Moritz D Lürig, Seth Donoughe, Erik I Svensson, Arthur Porto, and Masahito Tsuboi. Computer vision, machine learning, and the promise of phenomics in ecology and evolutionary biology. *Frontiers in Ecology and Evolution*, 9:642774, 2021.

Anniina Mattila, Chris Jiggins, and Ian Warren. University of Helsinki butterfly collection - Anniina Mattila bred specimens, February 2019. URL https://doi.org/10.5281/zenodo.2555086.

Joana I. Meier, Patricio Salazar, Gabriela Montejo-Kovacevich, Ian Warren, and Chris Jggins. Cambridge butterfly wing collection - Patricio Salazar PhD wild specimens batch 3, October 2020. URL https://doi.org/10.5281/zenodo.4153502.

Jason G Mezey and David Houle. The dimensionality of genetic variation for wing shape in drosophila melanogaster. *Evolution*, 59(5):1027–1038, 2005.

Francois Michonneau, Joseph W. Brown, and David J. Winter. rotl: an r package to interact with the open tree of life data. *Methods in Ecology and Evolution*, 7(12):1476–1481, 2016. doi: 10.1111/2041-210X.12593.

Gabriela Montejo-Kovacevich, Chris Jiggins, and Ian Warren. Cambridge butterfly wing collection batch 1- version 2, May 2019a. URL https://doi.org/10.5281/zenodo.3082688.

Gabriela Montejo-Kovacevich, Chris Jiggins, and Ian Warren. Cambridge butterfly wing collection batch 2, May 2019b. URL https://doi.org/10.5281/zenodo.2677821.

Gabriela Montejo-Kovacevich, Chris Jiggins, and Ian Warren. Cambridge butterfly wing collection batch 4, May 2019c. URL https://doi.org/10.5281/zenodo.2682669.

Gabriela Montejo-Kovacevich, Chris Jiggins, Ian Warren, Camilo Salazar, Marianne Elias, Imogen Gavins, Eva Wiltshire, Stephen Montgomery, and Owen McMillan. Cambridge and collaborators butterfly wing collection batch 10, May 2019d. URL https://doi.org/10.5281/zenodo.2813153.

Gabriela Montejo-Kovacevich, Chris Jiggins, Ian Warren, and Eva Wiltshire. Cambridge butterfly wing collection batch 7, May 2019e. URL https://doi.org/10.5281/zenodo.2702457.

Gabriela Montejo-Kovacevich, Chris Jiggins, Ian Warren, and Eva Wiltshire. Cambridge butterfly wing collection batch 8, May 2019f. URL https://doi.org/10.5281/zenodo.2707828.

Gabriela Montejo-Kovacevich, Chris Jiggins, Ian Warren, and Eva Wiltshire. Cambridge butterfly wing collection batch 5, May 2019g. URL https://doi.org/10.5281/zenodo.2684906.

Gabriela Montejo-Kovacevich, Chris Jiggins, Ian Warren, and Eva Wiltshire. Cambridge butterfly wing collection batch 6, May 2019h. URL https://doi.org/10.5281/zenodo.2686762.

Gabriela Montejo-Kovacevich, Chris Jiggins, Ian Warren, Eva Wiltshire, and Imogen Gavins. Cambridge butterfly wing collection batch 9, May 2019i. URL https://doi.org/10.5281/zenodo.2714333.

Gabriela Montejo-Kovacevich, Eva van der Heijden, and Chris Jiggins. Cambridge butterfly collection - GMK Broods Ikiam 2018, November 2020a. URL https://doi.org/10.5281/zenodo.4291095.

Gabriela Montejo-Kovacevich, Eva van der Heijden, Nicola Nadeau, and Chris Jiggins. Cambridge butterfly wing collection batch 10, November 2020b. URL https://doi.org/10.5281/zenodo.4289223.

Gabriela Montejo-Kovacevich, Quentin Paynter, and Amin Ghane. Heliconius erato cyrbia, Cook Islands (New Zealand) 2016, 2019, 2021, September 2021. URL https://doi.org/10.5281/zenodo.5526257.

Samuel G Müller and Frank Hutter. Trivialaugment: Tuning-free yet state-of-the-art data augmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 774–782, 2021.

Meike Nauta, Ron Van Bree, and Christin Seifert. Neural prototype trees for interpretable fine-grained image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14933–14943, 2021.

Meike Nauta, Jörg Schlötterer, Maurice van Keulen, and Christin Seifert. Pip-net: Patch-based intuitive prototypes for interpretable image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2744–2753, 2023.

Maureen A O'Leary and Seth Kaufman. Morphobank: phylophenomics in the "cloud". *Cladistics*, 27(5):529–537, 2011.

Dipanjyoti Paul, Arpita Chowdhury, Xinqi Xiong, Feng-Ju Chang, David Carlyn, Samuel Stevens, Kaiya Provost, Anuj Karpatne, Bryan Carstens, Daniel Rubenstein, et al. A simple interpretable transformer for fine-grained image classification and analysis. *arXiv preprint arXiv:2311.04157*, 2023.

Matthew W Pennell, Richard G FitzJohn, William K Cornwell, and Luke J Harmon. Model adequacy and the macroevolution of angiosperm functional traits. *The American Naturalist*, 186(2): E33–E50, 2015.

Erika Pinheiro de Castro, Christopher Jiggins, Karina Lucas da Silva-Brandŏ0e3o, Andre Victor Lucci Freitas, Marcio Zikan Cardoso, Eva Van Der Heijden, Joana Meier, and Ian Warren. Brazilian Butterflies Collected December 2020 to January 2021, February 2022. URL https://doi.org/10.5281/zenodo.5561246.

MARTin J RAMirez, Jonathan A Coddington, Wayne P Maddison, Peter E Midford, Lorenzo Prendini, Jeremy Miller, Charles E Griswold, Gustavo Hormiga, Petra Sierwald, Nikolaj Scharff, et al. Linking of digital images to phylogenetic data matrices using a morphological ontology. *Systematic Biology*, 56(2):283–294, 2007.

Benjamin Redelings, Luna Luisa Sanchez Reyes, Karen A. Cranston, Jim Allman, Mark T. Holder, and Emily Jane McTavish. Open tree of life synthetic tree, 2019. URL https://doi.org/10.5281/zenodo.3937741.

Dawid Rymarczyk, Łukasz Struski, Jacek Tabor, and Bartosz Zieliński. Protopshare: Prototypical parts sharing for similarity discovery in interpretable image classification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 1420–1430, 2021.

Dawid Rymarczyk, Łukasz Struski, Michał Górszczak, Koryna Lewandowska, Jacek Tabor, and Bartosz Zieliński. Interpretable image classification with differentiable prototypes assignment. In *European Conference on Computer Vision*, pp. 351–368. Springer, 2022.

Camilo Salazar, Gabriela Montejo-Kovacevich, Chris Jiggins, Ian Warren, and Imogen Gavins. Camilo Salazar and Cambridge butterfly wing collection batch 1, May 2019a. URL https://doi.org/10.5281/zenodo.2735056.

Patricio Salazar, Gabriela Montejo-Kovacevich, Ian Warren, and Chris Jiggins. Cambridge butterfly wing collection - Patricio Salazar PhD wild and bred specimens batch 1, December 2018. URL https://doi.org/10.5281/zenodo.1748277.

Patricio Salazar, Gabriela Montejo-Kovacevich, Ian Warren, and Chris Jiggins. Cambridge butterfly wing collection - Patricio Salazar PhD wild and bred specimens batch 2, January 2019b. URL https://doi.org/10.5281/zenodo.2548678.

Patricio A. Salazar, Nicola Nadeau, Gabriela Montejo-Kovacevich, and Chris Jiggins. Sheffield butterfly wing collection - Patricio Salazar, Nicola Nadeau, Ikiam broods batch 1 and 2, November 2020. URL https://doi.org/10.5281/zenodo.4288311.

Paul C Sereno. Logical basis for morphological characters in phylogenetics. *Cladistics*, 23(6): 565–587, 2007.

Thalles Silva and Adín Ramírez Rivera. Representation learning via consistent assignment of views to clusters. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, pp. 987–994, 2022.

Tiago R Simões, Michael W Caldwell, Alessandro Palci, and Randall L Nydam. Giant taxon-character matrices: quality of character constructions remains critical regardless of size. *Cladistics*, 33(2):198–219, 2017.

Randal A Singer, Kevin J Love, and Lawrence M Page. A survey of digitized data from us fish collections in the idigbio data aggregator. *PloS one*, 13(12):e0207636, 2018.

Fariborz Taherkhani, Hadi Kazemi, Ali Dabouei, Jeremy Dawson, and Nasser M Nasrabadi. A weakly supervised fine label classifier enhanced by coarse supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6459–6468, 2019.

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8769–8778, 2018.

Grant Van Horn, Elijah Cole, Sara Beery, Kimberly Wilber, Serge Belongie, and Oisin Mac Aodha. Benchmarking representation learning for natural world image collections. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12884–12893, 2021.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

Jiaqi Wang, Huafeng Liu, Xinyue Wang, and Liping Jing. Interpretable image recognition by constructing transparent embedding space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 895–904, 2021.

Jiayun Wang, Yubei Chen, Rudrasis Chakraborty, and Stella X Yu. Orthogonal convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11505–11515, 2020.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pp. 9929–9939. PMLR, 2020.

Ian Warren and Chris Jiggins. Miscellaneous Heliconius wing photographs (2001-2019) Part 2, February 2019a. URL `https://doi.org/10.5281/zenodo.2553501`.

Ian Warren and Chris Jiggins. Miscellaneous Heliconius wing photographs (2001-2019) Part 1, February 2019b. URL `https://doi.org/10.5281/zenodo.2552371`.

Ian Warren and Chris Jiggins. Miscellaneous Heliconius wing photographs (2001-2019) Part 3, February 2019c. URL `https://doi.org/10.5281/zenodo.2553977`.

Zhicheng Yan, Hao Zhang, Robinson Piramuthu, Vignesh Jagadeesh, Dennis DeCoste, Wei Di, and Yizhou Yu. Hd-cnn: hierarchical deep convolutional neural networks for large scale visual recognition. In *Proceedings of the IEEE international conference on computer vision*, pp. 2740–2748, 2015.

APPENDIX

TABLE OF CONTENTS

A Additional Biological Background **16**

B Ablation of Over-specificity Loss Trade-off Hyperparameter **17**

C Ablation of Number of Prototypes **17**

D Ablation of Individual Losses **17**

E Consistency of Classification Performance Over Multiple Runs **18**

F Implementation Details **18**

G Post-hoc Thresholding to Identify Over-specific Prototypes **20**

H Visual Comparison of a Over-specific and a Non-over-specific prototype **20**

I Performance of HPnet with high resolution feature maps **21**

J Additional Visualizations of the Hierarchical Prototypes Discovered by HComP-Net **21**

K Additional Top-K Visualizations of HComP-Net Prototypes **21**

L Reproducibility **23**

M Limitations of Our Work **24**

A ADDITIONAL BIOLOGICAL BACKGROUND

One of the first steps in any study of evolutionary morphology is *character construction* - the process of deciding which measurements will be taken of organismal variation that are replicable and meaningful for the underlying biology, and how these traits should be represented numerically (Mezey & Houle, 2005). For phylogenetic studies, researchers typically attempt to identify *synapomorphies* – versions of the traits that are shared by two or more species, are inherited from their most recent common ancestor, and may have evolved along the phylogeny branch. The difficulty with the traditional character construction process is that humans often measure traits in a way that is inconsistent and difficult to reproduce, and can neglect shared features that may represent synapomorphies, but defy easy quantification. To address the problem of human inconsistency, PhyloNN (Elhamod et al., 2023) and Phylo-Diffusion (Khurana et al., 2024) took a knowledge-guided machine learning (KGML) (Karpatne et al., 2024) approach to character construction, by giving their neural networks knowledge about the biological process they were interested in studying (in their case, phylogenetic history), and specifically optimizing their models to find embedded features (analogous to biological traits) that are predictive of that process. To address the problem of visual irreproducibility, RAMirez et al. (2007) suggested photographing the local structures where the empirical traits vary and linking the images to written descriptions of the traits. In this paper, we take influence from both approaches. We extend the hierarchical prototype approach from Hase et al. (2019) to better reflect phylogeny, similar in theory to the way PhyloNN (Elhamod et al., 2023) and Phylo-Diffusion (Khurana et al., 2024) learned embeddings that reflect phylogeny. Using prototypes, however, we

enforce local visual interpretability similar to how researchers may use "type-specimens" to define prototypical definitions of particular character states.

Specifically, our method is about finding synapomorphies–shared derived features unique to a particular group of species that share a common ancestor in the phylogeny (referred to as clade). While such features may bear similarities to convergent phenotypes in other clades, our goal is not to identify features that exhibit convergence. It is typical for phylogenetic studies to specifically avoid features that exhibit high levels of convergence, as they can lend support for erroneous phylogenetic relationships. Identifying convergence requires additional information such as shared habitat, niche, diet, or behavior, which is not incorporated in our work.

B Ablation of Over-specificity Loss Trade-off Hyperparameter

We have provided an ablation for the over-specificity loss trade-off hyperparameter (λ_{ovsp}) in Table 4. We can observe that increasing the weight of over-specificity loss reduces the model's classification performance, as the model struggles to find any commonality, especially at internal nodes where the number of leaf descendant species is large and quite diverse. It is natural that species that are diverse and distantly related may share fewer characteristics with each other, in comparison to a set of species that diverged more recently from a common ancestor (Harmon et al., 2010; Pennell et al., 2015). Therefore, forcing the model to learn common traits with a strong $\mathcal{L}_{ovsp}$ constraint can cause the model to perform badly in terms of accuracy.

Table 4: Ablation of over-specificity loss trade-off hyperparameter (λ_{ovsp}). Done on Bird dataset.

λ_{ovsp}	Part purity	Part purity with mask applied	% masked	% Accuracy
w/o $\mathcal{L}_{ovsp}$	0.68 ± 0.22	0.75 ± 0.17	21.42%	58.32
0.05	$\mathbf{0.72 \pm 0.19}$	$\mathbf{0.77 \pm 0.16}$	16.53%	70.01
0.1	0.71 ± 0.18	0.74 ± 0.16	11.31%	**70.97**
0.5	0.71 ± 0.19	0.72 ± 0.18	4.2%	68.23
1.0	0.70 ± 0.19	0.70 ± 0.2	2.13%	62.68
2.0	0.69 ± 0.19	0.69 ± 0.19	0.55%	53.16

C Ablation of Number of Prototypes

In Table 5, we vary the number of prototypes per child β for a node to see the impact on the model's performance. We note that while the accuracy increases marginally with increasing the number of prototypes per child (β) from 10 to 15, it does not affect the performance of the model significantly. Therefore, we continue to work with $\beta = 10$ for all of our experiments.

Table 5: Ablation of the number of prototypes per child for a node (β). Done on Bird dataset.

Number of Prototypes (β)	% Accuracy
10	70.01
15	**70.92**
20	69.99

D Ablation of Individual Losses

In Table 6, we perform an ablation of the various loss terms used in our methodology. As it can be observed, the removal of $\mathcal{L}_{ovsp}$ and $\mathcal{L}_{disc}$ degrades performance in terms of both semantic consistency (part purity) and accuracy. On the other hand, the removal of self-supervised contrastive loss $\mathcal{L}_{SS}$ improves accuracy but at the cost of drastically decreasing the semantic consistency.

Table 6: Ablation of individual losses. Done on Bird dataset.

Model	Part purity	Part purity with mask applied	% masked	% Accuracy
HComP-Net	0.72 ± 0.19	0.77 ± 0.16	16.53%	70.01
HComP-Net w/o $\mathcal{L}_{ovsp}$	0.68 ± 0.22	0.75 ± 0.17	21.42%	58.32
HComP-Net w/o $\mathcal{L}_{disc}$	0.69 ± 0.19	0.72 ± 0.17	10.95%	65.99
HComP-Net w/o $\mathcal{L}_{SS}$	0.53 ± 0.18	0.57 ± 0.15	8.36%	81.62

E Consistency of Classification Performance Over Multiple Runs

We trained the model using five distinct random weight initializations. The results showed that the model's fine-grained accuracy averaged 70.63% with a standard deviation of 0.18% on the Bird dataset.

F Implementation Details

We have included all the source code and dataset along with the comprehensive instructions to reproduce the results, in the supplementary material (.zip file).

Model hyper-parameters: We build HComP-Net on top of a ConvNeXt-tiny (Liu et al., 2022) architecture as the backbone feature extractor. We have modified the stride of the max pooling layers of later stages of the backbone from 2 to 1, similar to PIP-Net, such that the backbone produces feature maps of increased height and width, in order to get more fine-grained prototype score maps. We implement and experiment with our method on ConvNeXt-tiny backbones with 26×26 feature maps. The length of prototype vectors C is 768. The weights ϕ at every node n of HComP-Net are constrained to be non-negative by the use of the ReLU activation function (Agarap, 2018). Further, the prototype activation nodes are connected with non-negative weights only to their respective child classes in W while their weights to other classes are made zero and non-trainable. For hyper-parameter tuning HPnet for Spider and Turtle datasets we tried various prototype vector lengths such as 128, 512, 1024. We also increased the learning rate by a factor of 10. We also reduced push operation from once every 5 epochs to only done at the end of training. We found prototype length of 128 with original learning and push operation done at the end of training to perform the best.

Training details: All models were trained with images resized and appropriately padded to 224×224 pixel resolution and augmented using TrivialAugment (Müller & Hutter, 2021) for contrastive learning. The prototypes are pretrained with self-supervised learning similar to PIP-Net for 10 epochs, following which the model is trained with the entire set of loss functions for 60 epochs. We use a batch size of 256 for the Bird dataset and 64 for the Butterfly and Fish dataset. The masking module is trained in parallel, and its training is continued for 15 additional epochs after the training of the rest of the model is completed. The trade-off hyper-parameters for the loss functions are set to be $\lambda_{CE} = 2; \lambda_A = 5; \lambda_T = 2; \lambda_{ovsp} = 0.05; \lambda_{disc} = 0.1; \lambda_{orth} = 0.1; \lambda_{mask} = 2.0; \lambda_{L1} = 0.5$. λ_{CE}, λ_T and λ_A were borrowed from PIP-Net (Nauta et al., 2023). Ablations to arrive at suitable λ_{ovsp} is provided in Table 4. λ_{disc} and λ_{orth} were chosen empirically and found to work well on all three datasets. Experiment on unseen species was done by leaving out certain classes from the datasets, so that they are not considered during training.

Dataset and Phylogeny Details: Dataset statistics and phylogeny statistics are provided in Table 8 and Table 7 respectively. Bird dataset is created by choosing 190 species from CUB-200-2011 [1] (Wah et al., 2011) dataset, which were part of the phylogeny. Background from all images was filtered using the associated segmentation metadata (Farrell, 2024). For Fish [2] dataset, we followed the exact same preprocessing steps as outlined in PhyloNN (Elhamod et al., 2023). We obtain the Spider and Turtle dataset from iNaturalist by filtering for the *Araneae* order and the *Testudines* order respectively. We manually observe that images from iNaturalist are noisy and contain undesired background artifacts which may deter our models from learning attributes on the organism

[1] License: CC BY

[2] License: CC BY-NC

body. To account for this, we use GroundingDINO Liu et al. (2023) to detect and crop into 'complete turtle' and 'complete spider' with a threshold confidence of 50%. We filter images in which GroundingDINO cannot detect the organisms with sufficient confidence. For Butterfly dataset we considered each subspecies as an individual class and considered only the subspecies of genus Heliconius from the Heliconius Collection (Cambridge Butterfly)[3] (Lawrence & Campolongo, 2024). There is substantial variation among subspecies of Heliconius species. Furthermore, we balanced the dataset by filtering out the subspecies that did not have 20 or more images. We also sampled a subset of 100 images from each subspecies that had more than 100 images.

Compute Resources: The models for the Bird dataset were trained on two NVIDIA A100 GPUs with 80GB of RAM each. Butterfly and Fish models were trained on a single A100 GPU. As a rough estimate, the execution time for the training model on the Bird dataset is around 2.5 hours. For Butterfly and Fish datasets, the training is completed in under 1 hour. We used a single A100 GPU during the inference stage for all other analyses.

Table 7: High-level statistics of the phylogenies used for different datasets.

Phylogeny	# Internal nodes	Max-depth	Min-depth
Bird	184	25	3
Butterfly	13	5	2
Fish	20	11	2
Spider	53	17	2
Turtle	38	12	1

Table 8: Dataset statistics (# train and validation images).

Dataset	# Classes	Train set	Validation set
Bird	190	5695	5512
Butterfly	30	1418	358
Fish	38	4140	1294
Spider	113	26784	991
Turtle	41	9951	345

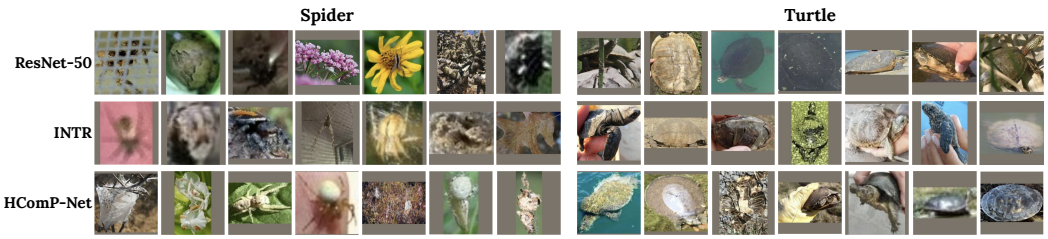


Figure 7: Sample of images misclassified by ResNet-50, INTR and HComP-Net models on Spider and Turtle dataset. We observe that the dataset is quite noisy with diverse backgrounds along with obscured and blurred images making it a challenging task to identify and interpret commonalities.

[3]Note that this dataset is a compilation of images from 25 Zenodo records by the Butterfly Genetics Group at Cambridge University, licensed under Creative Commons Attribution 4.0 International (Montejo-Kovacevich et al., 2020b; Salazar et al., 2020; Montejo-Kovacevich et al., 2019b; Jiggins et al., 2019; Montejo-Kovacevich et al., 2019c;g; Warren & Jiggins, 2019b;c; Montejo-Kovacevich et al., 2019h; Jiggins & Warren, 2019a;b; Meier et al., 2020; Montejo-Kovacevich et al., 2019a;d; Salazar et al., 2018; Montejo-Kovacevich et al., 2019e; Salazar et al., 2019b; Pinheiro de Castro et al., 2022; Montejo-Kovacevich et al., 2019f;i; 2020a; 2021; Warren & Jiggins, 2019a; Salazar et al., 2019a; Mattila et al., 2019).

Table 9: Part purity with post-hoc thresholding approach. Done on Bird dataset.

Threshold	Part purity with mask applied	% masked
0.2	0.74 ± 0.28	12.28%
0.3	0.75 ± 0.27	13.47%
0.4	0.76 ± 0.26	14.97%
0.5	0.77 ± 0.15	16.66%
0.6	0.77 ± 0.26	17.43%

G POST-HOC THRESHOLDING TO IDENTIFY OVER-SPECIFIC PROTOTYPES

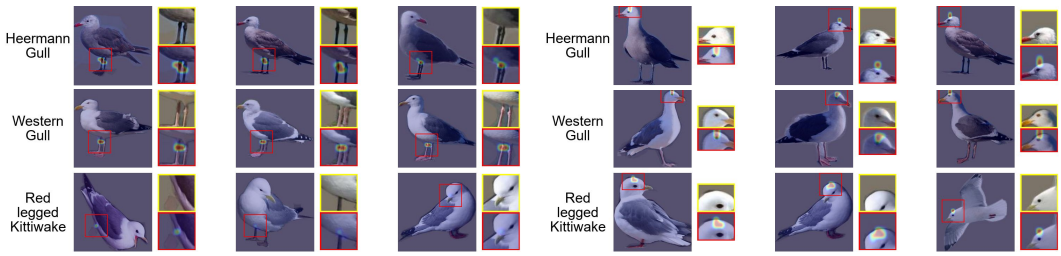
An alternative approach to learning the masking module is to calculate the over-specificity score for each prototype on the test set after training the model. We calculate the over-specificity scores for the prototypes of a trained model as follows,

$$\mathcal{O}_i = -\prod_{d=1}^{D_i} \frac{1}{\text{top}_k} \sum_{i=1}^{\text{top}_k} (\mathbf{g_i}) \tag{9}$$

For a given prototype, we choose the top_k images with the highest prototype scores from each leaf descendant. After taking mean of the top_k prototype score, we multiply the values from each descendant to arrive at the over-specificity score for the particular prototype. Subsequently, we choose a threshold to determine which prototypes are over-specific. We provide the results of post-hoc thresholding approach that can also be used to identify over-specific prototypes in Table 9. While we can note that this approach can also be effective, validating the threshold particularly in scenarios where there is no part annotations available (such as part location annotation of CUB-200-2011) can be an arduous task. In such cases, directly identifying over-specific prototypes as part of the training through the masking module can be the more feasible option.

H VISUAL COMPARISON OF A OVER-SPECIFIC AND A NON-OVER-SPECIFIC PROTOTYPE

We do a visual comparison of a prototype that has been identified as over-specific by the masking module and a prototype that is not identified as over-specific in Figure 8. As it can be observed in Figure 8(a), the Red-Legged Kittiwake has legs that are shorter in comparison to other species of its clade - Heerman Gull and Western Gull. Therefore, the prototype is identified as over-specific, as long legs are not common to all three species. On the other hand, in Figure 8(b), the prototype has been identified as non-over-specific because all three species share white-colored crowns. Prototype from Figure 8(a) has very low activation for Red Legged Kittiwake and also has poor part purity since it does not highlight the same part of the bird in the images of Red-legged Kittiwake.



(a) Masked prototype (Over-specific) (b) Unmasked prototype (Non-over-specific)

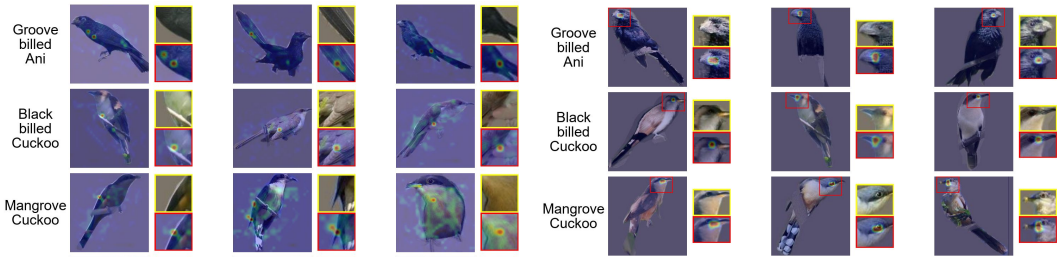
Figure 8: Comparison of over-specific (a) vs non-over-specific (b) prototype identified by masking module at the same internal node. Each row corresponds to top-3 closest image to the prototype from each species.

I PERFORMANCE OF HPNET WITH HIGH RESOLUTION FEATURE MAPS

We analyze the performance of HPnet with high-resolution feature maps in Table 10. We modified the backbone by removing the max pooling layers at the final stages of the model to produce a 28×28 feature map instead of the original 7×7 feature map. It can be observed that the accuracy and part purity do not improve with high-resolution feature maps. We also make a qualitative comparison between an HPnet and HComP-Net prototype with a higher resolution feature map in Figure 9, showing that part purity does not improve with high-resolution feature maps for HPnet.

Table 10: Performance of HPnet with higher resolution feature map (Feature map dimensions in parenthesis)

Model	% Accuracy	Part purity
HComP-Net (26×26)	70.01	0.77 ± 0.16
HPnet (7×7)	36.18	0.14 ± 0.09
HPnet (28×28)	20.68	0.14 ± 0.11



(a) HPnet prototype with 28×28 feature map (b) HComP-Net prototype with 26×26 feature map

Figure 9: Comparison of HPnet (28×28 feature map) (a) and HComP-Net (26×26 feature map) (b) prototype score maps. Although HPnet with a 28×28 feature map highlights a localized region in the image, the prototype highlights varying regions in each image. HComP-Net prototype visualization is more localized and is also consistent in the part it highlights.

J ADDITIONAL VISUALIZATIONS OF THE HIERARCHICAL PROTOTYPES DISCOVERED BY HCOMP-NET

We provide more visualizations of the hierarchical prototypes discovered by HComP-Net for Butterfly (Figures 10 and 11), Fish (Figure 12) and Spider and Turtle (Figure 13) datasets in this section. For ease of visualization, in each figure we visualize the prototypes learned over a small subtree from the phylogeny. The prototypes at the lowest level capture traits that are species-specific, whereas the prototypes at internal nodes capture the commonality between its descendant species. For Fish dataset, we have provided textual descriptions purely based on human interpretation for the traits that are captured by prototypes at different levels. For Butterfly, Spider and Turtle datasets, since the prototypes are capturing different patterns, assigning textual descriptions for them is not straightforward. Therefore, we refrain from providing any text description for the highlighted regions of the learned prototypes and leave it to the reader's interpretation.

K ADDITIONAL TOP-K VISUALIZATIONS OF HCOMP-NET PROTOTYPES

We provide additional top-K visualizations of the prototypes from Butterfly (Figures 14 to 17) and Fish (Figures 18 to 20) datasets, where every row corresponds to a descendant species and the columns corresponds to the top-K images from the species with the largest prototype activation scores. A requirement of a semantically meaningful prototype is that it should consistently highlight the same part of the organisms in various images, provided that the part is visible. We can see in the figures that the prototypes learned by HComP-Net consistently highlight the same part across all

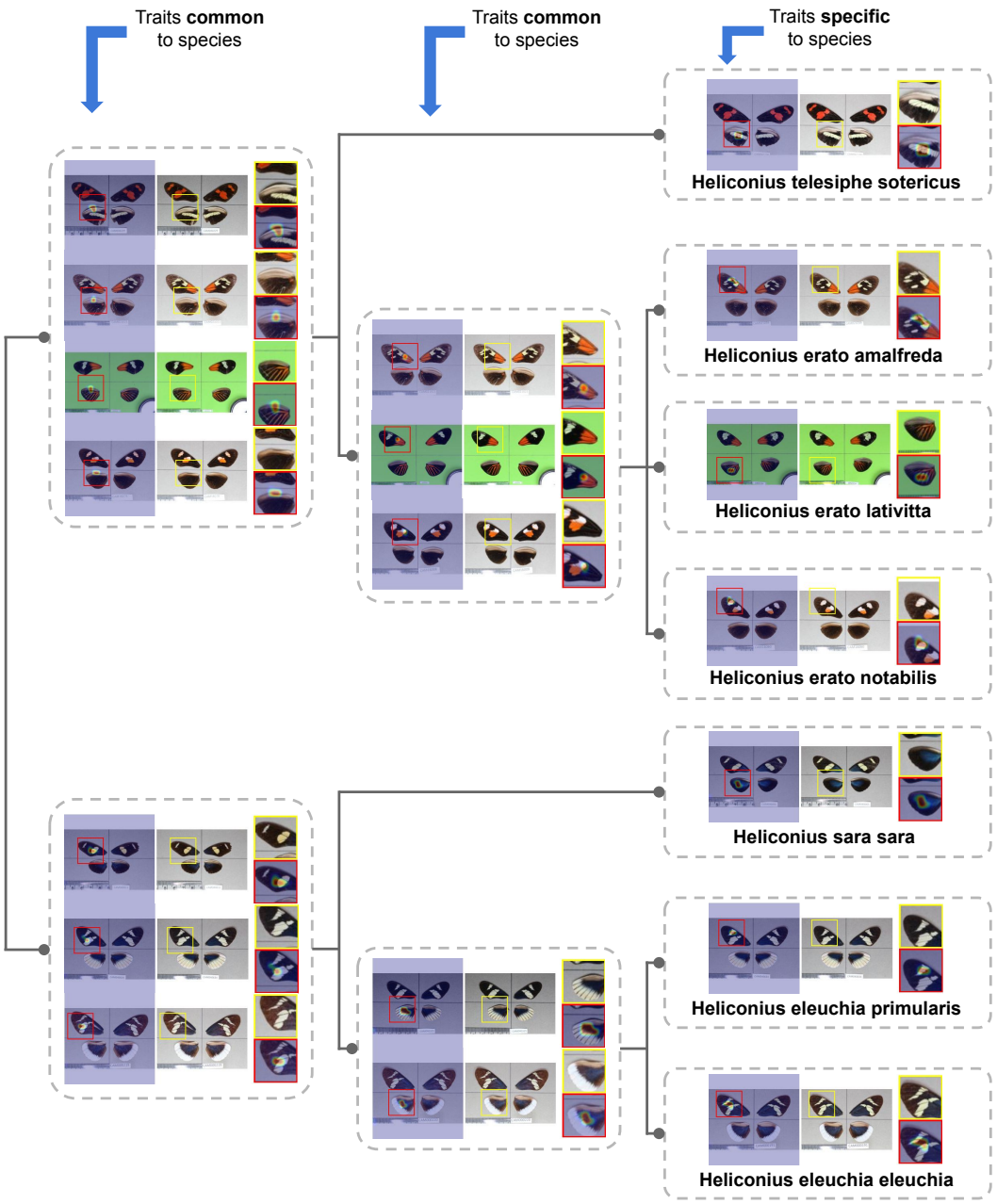


Figure 10: Visualizing the hierarchy of prototypes discovered by HComP-Net over three levels in the phylogeny of seven species from **Butterfly** dataset. For each prototype, we visualize one image from each of its leaf descendants. Therefore, for prototypes at the species level (rightmost column), we show only one image, whereas for prototypes at internal nodes, we show multiple images (equal to the number of leaf descendants). For each image, we show the zoomed-in view of the original image as well as the heatmap overlaid image in the region of the learned prototype. The prototypes appear to capture different butterfly wing patterns.

top-K images of a species, and across all descendant species. We additionally show that HComP-Net can find common traits at internal nodes with a varying number of descendant species, including 4 species (Figure 14), 5 species (Figures 15 and 16), and 10 species (Figure 17) of butterflies, and 5 species (Figure 18), 8 species (Figure 19) and 18 species (Figure 20) for fish. We also provide several top-k visualizations of prototypes learned for bird species in Figures 21 to 29. This shows

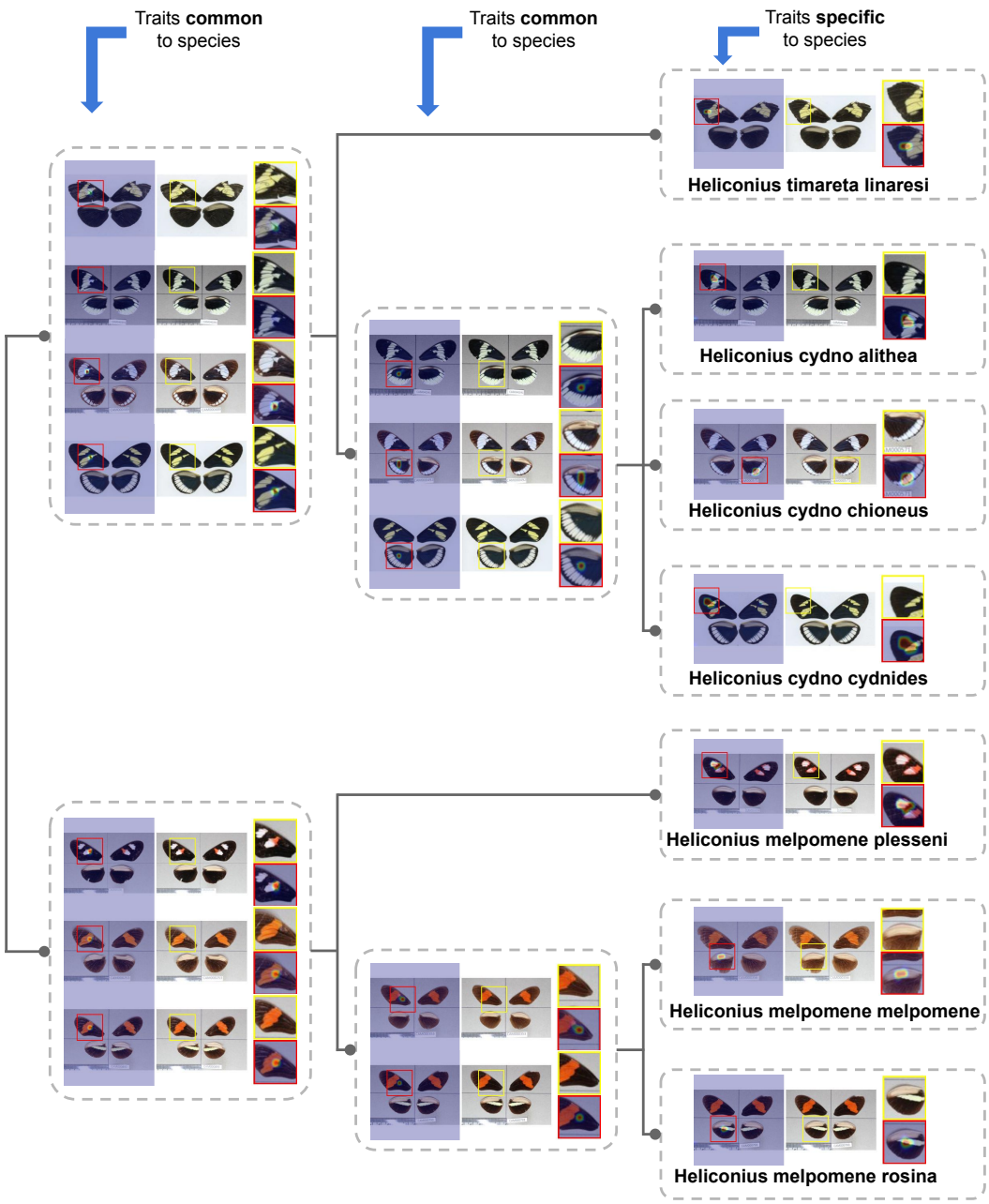


Figure 11: Visualizing the hierarchy of prototypes discovered by HComP-Net over three levels in the phylogeny of seven species from **Butterfly** dataset.

the ability of HComP-Net to discover common prototypes at internal nodes of the phylogenetic tree that consistently highlight the same regions in the descendant species images even when the number of descendants is large.

L Reproducibility

We have ensured the reproducibility of our work by providing a detailed set of resources. The source code used for training and evaluation, along with instructions for reproducing the experiments, is available via an anonymous GitHub link (https://anonymous.4open.science/r/HComP-Net-ICLR-14F6/). Comprehensive implementation details, including hyperparameters and compute resources,

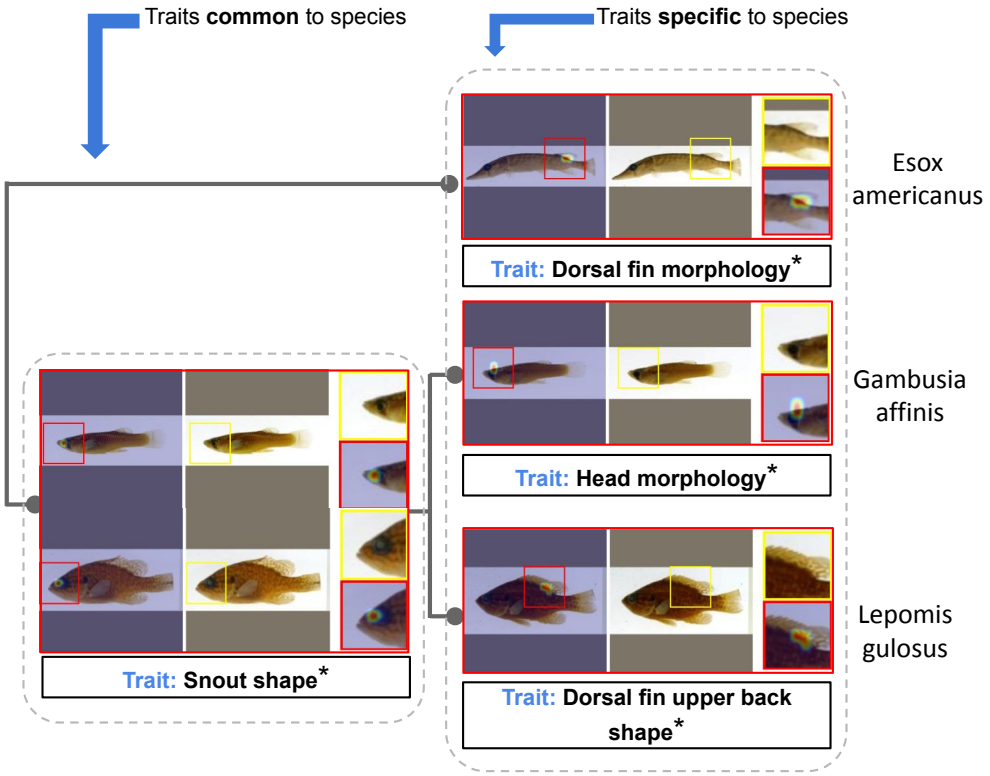


Figure 12: Visualizing the hierarchy of prototypes discovered by HComP-Net for a sub-tree with three species from **Fish** dataset. *Note that the textual descriptions of the hypothesized traits shown for every prototype are based on human interpretation.

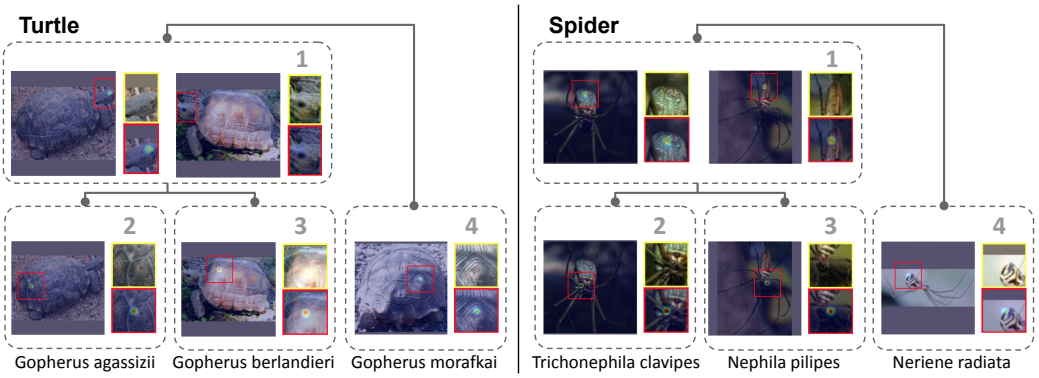


Figure 13: Visualizing the hierarchy of prototypes discovered by HComP-Net for a sub-tree with three species from **Turtle** and **Spider** datasets.

are described in Appendix F. Additionally, we have cited all dataset sources and outlined the data processing steps in Appendix F to facilitate accurate replication of our experiments.

M Limitations of Our Work

A fundamental challenge of every prototype-based interpretability method (including ours) is the difficulty in associating a semantic interpretation with the underlying visual concept of a prototype. While some prototypes can be interpreted easily based on visual inspection of prototype activation

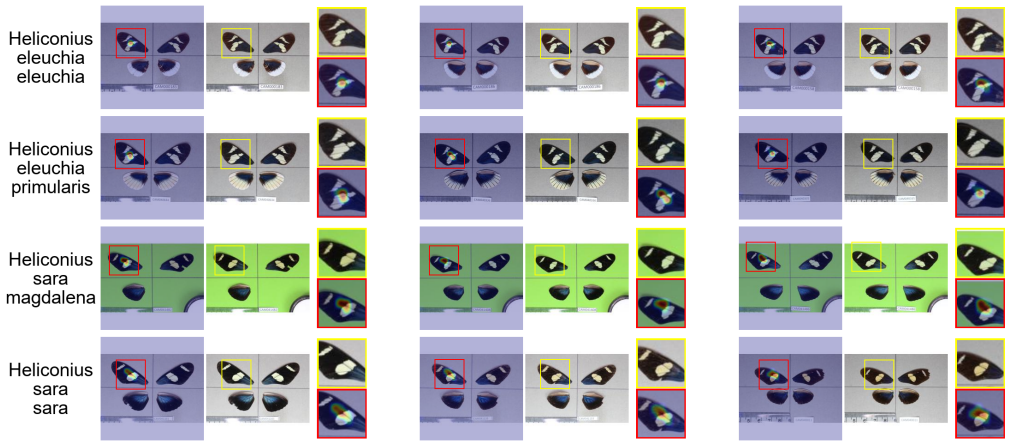


Figure 14: Top-K visualization of a prototype finding commonality between four species of butterfly sharing a common ancestor. Each row represents the top 3 images from the respective species. For each image, we show the zoomed-in view of the original image as well as the heatmap overlaid image.

maps, other prototypes are harder to interpret and require additional domain expertise of biologists. Also, while we have considered large phylogenies as that of the 190 species from the CUB dataset, it may still not be representative of all bird species. This limited scope may cause our method to identify apparent homologous evolutionary traits that could differ with the inclusion of more species into the phylogeny. Therefore, our method can be seen as a system that generates potential hypotheses about evolutionary traits discovered in the form of hierarchical prototypes.

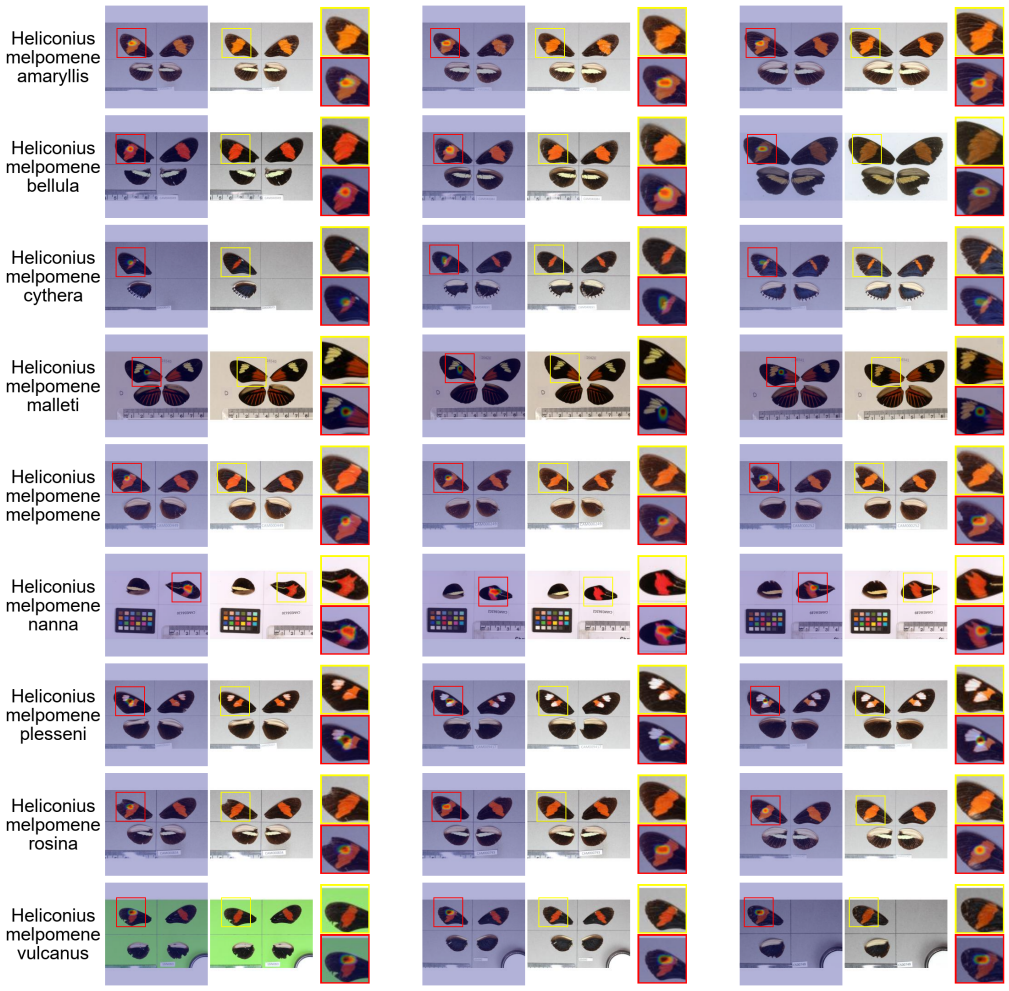


Figure 15: Top-K visualization of a prototype finding commonality between nine species of butterfly sharing a common ancestor. Each row represents the top 3 images from the respective species. For each image, we show the zoomed-in view of the original image as well as the heatmap overlaid image.

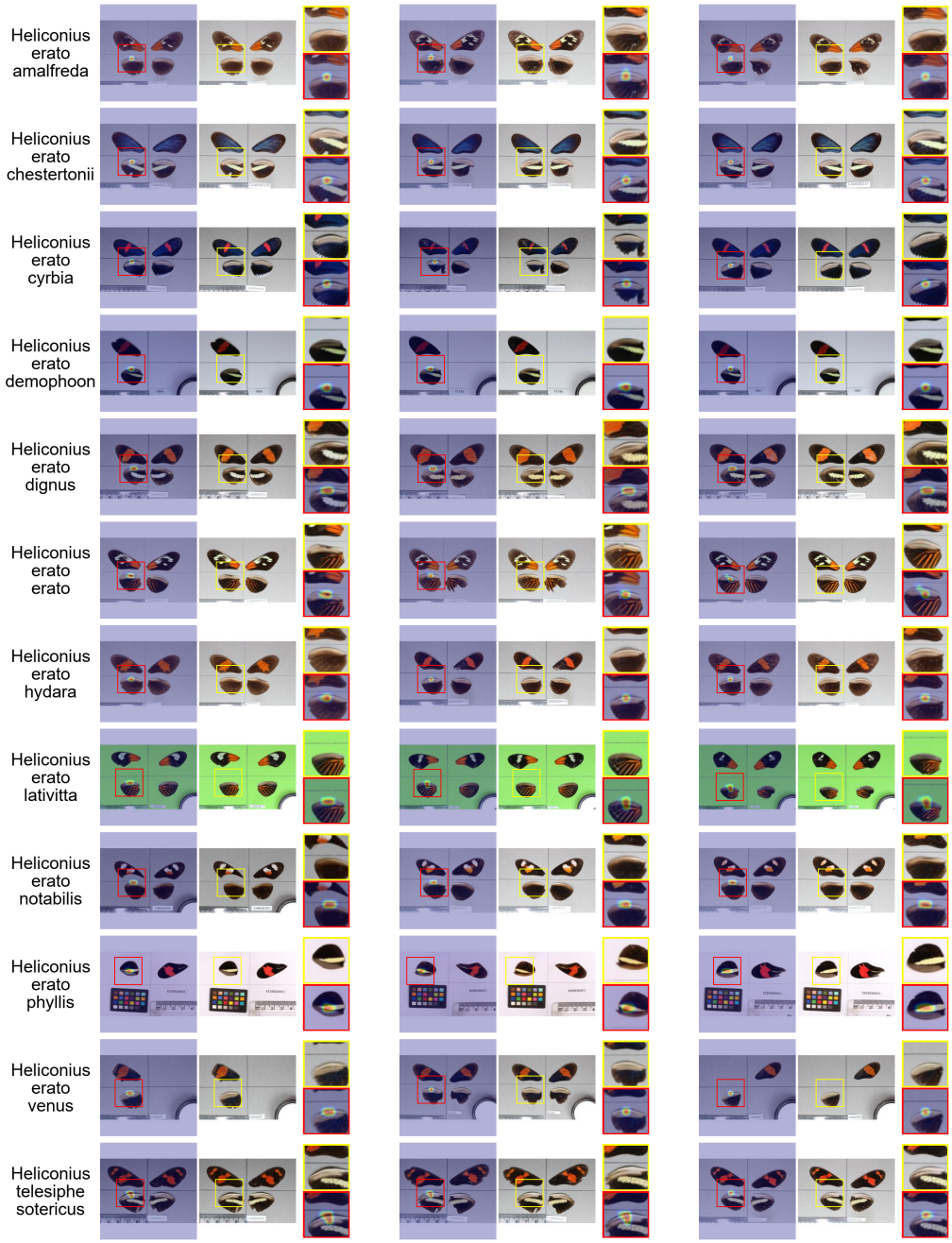


Figure 16: Top-K visualization of a prototype finding commonality between twelve species of butterfly sharing a common ancestor. Each row represents the top 3 images from the respective species. For each image, we show the zoomed-in view of the original image as well as the heatmap overlaid image.

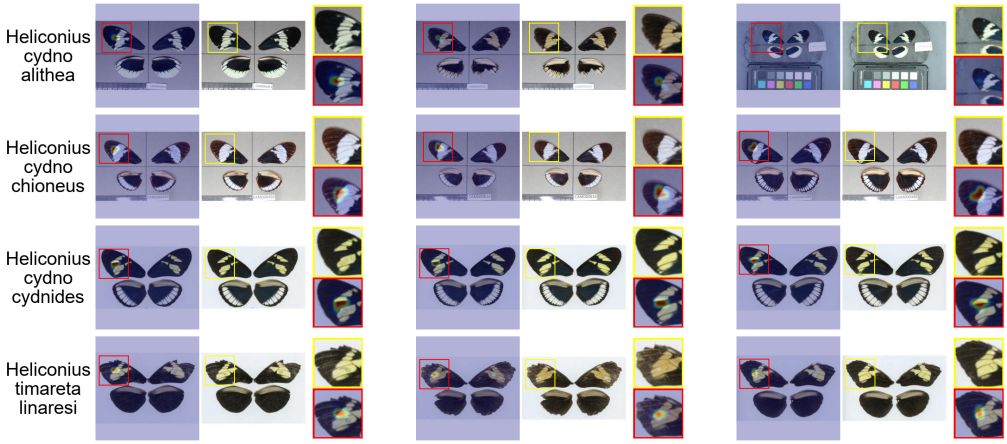


Figure 17: Top-K visualization of a prototype finding commonality between four species of butterfly sharing a common ancestor. Each row represents the top 3 images from the respective species. For each image, we show the zoomed-in view of the original image as well as the heatmap overlaid image.

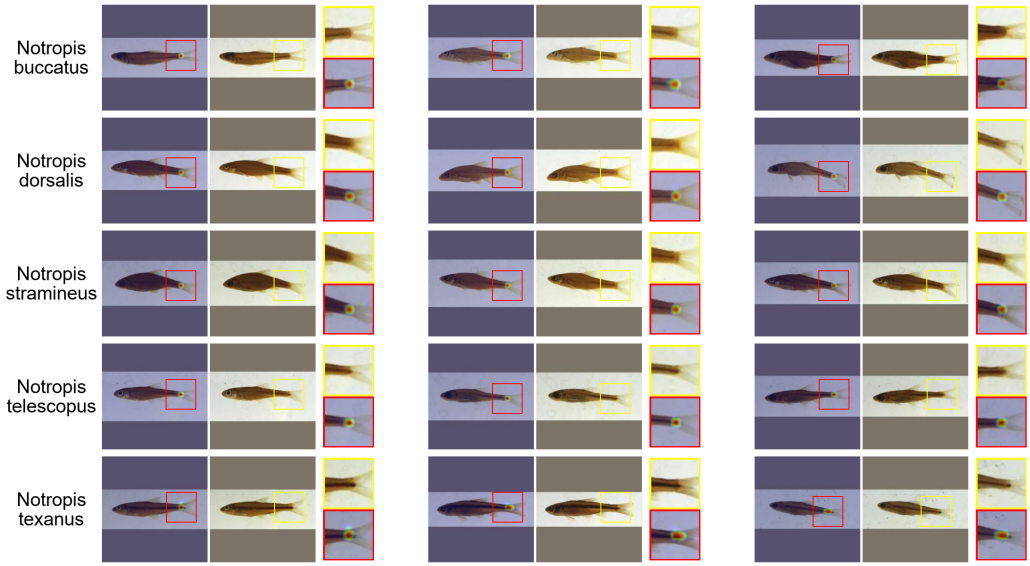


Figure 18: Top-K visualization of a prototype finding commonality between five species of fish sharing a common ancestor. Each row represents the top 3 images from the respective species. For each image, we show the zoomed-in view of the original image as well as the heatmap overlaid image.

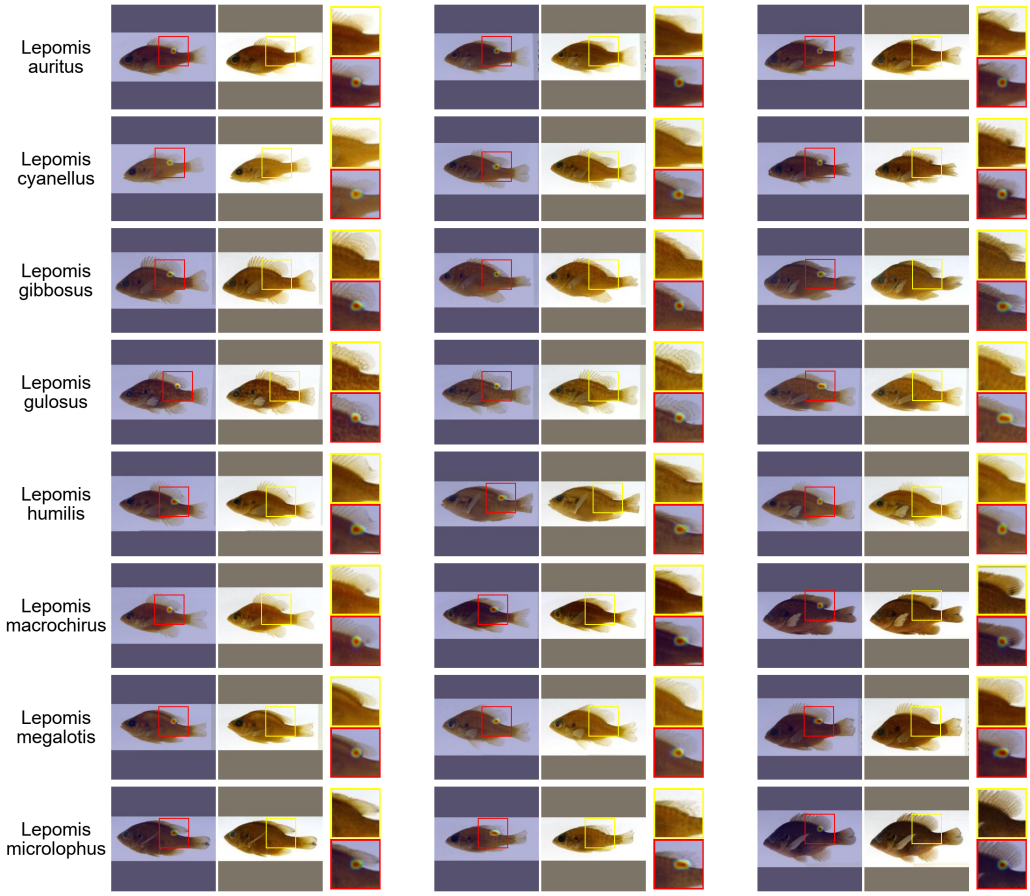


Figure 19: Top-K visualization of a prototype finding commonality between eight species of fish sharing a common ancestor. Each row represents the top 3 images from the respective species. For each image, we show the zoomed-in view of the original image as well as the heatmap overlaid image.

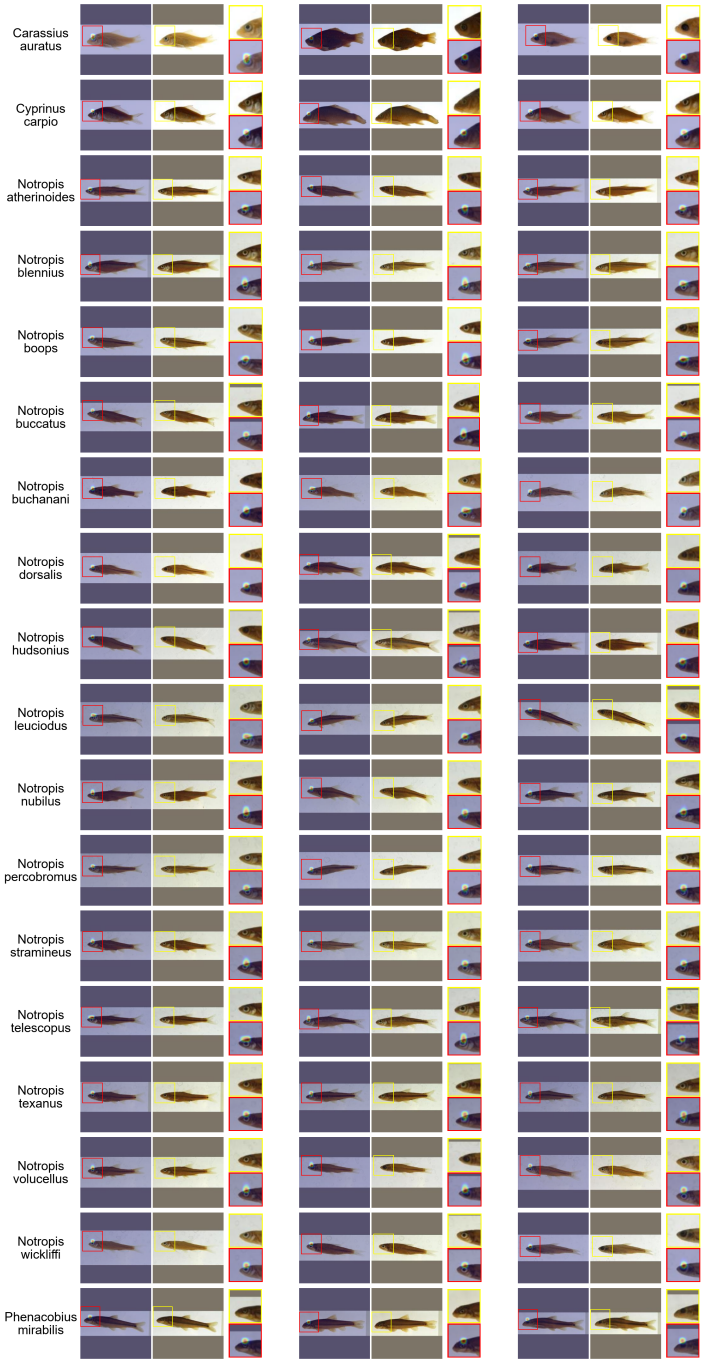


Figure 20: Top-K visualization of a prototype finding commonality between eighteen species of fish sharing a common ancestor. Each row represents the top 3 images from the respective species. For each image, we show the zoomed-in view of the original image as well as the heatmap overlaid image.

Figure 21: Top-K visualization of a prototype finding commonality between seven species of birds sharing a common ancestor. Each row represents the top 3 images from the respective species. For each image, we show the zoomed-in view of the original image as well as the heatmap overlaid image.

Figure 22: Top-K visualization of a prototype finding commonality between eight species of birds sharing a common ancestor. Each row represents the top 3 images from the respective species. For each image, we show the zoomed-in view of the original image as well as the heatmap overlaid image.

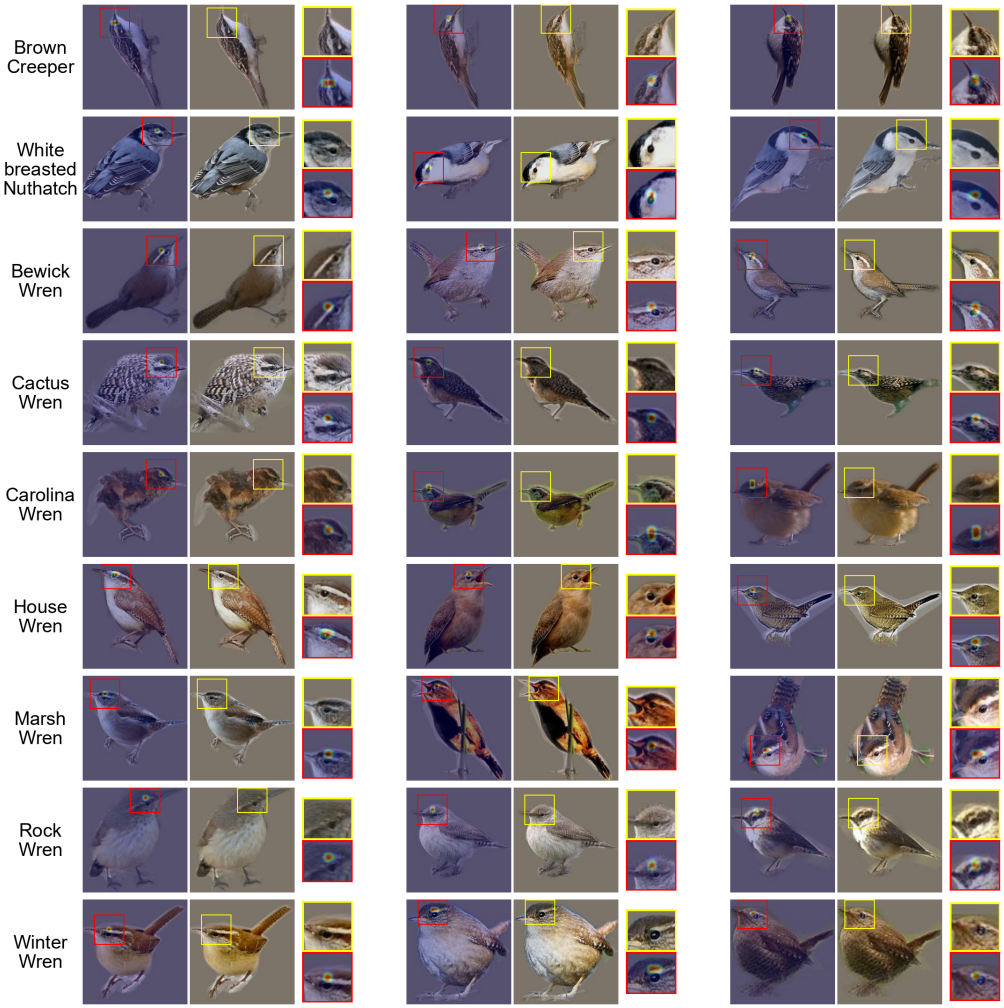


Figure 23: Top-K visualization of a prototype finding commonality between nine species of birds sharing a common ancestor. Each row represents the top 3 images from the respective species. For each image, we show the zoomed-in view of the original image as well as the heatmap overlaid image.

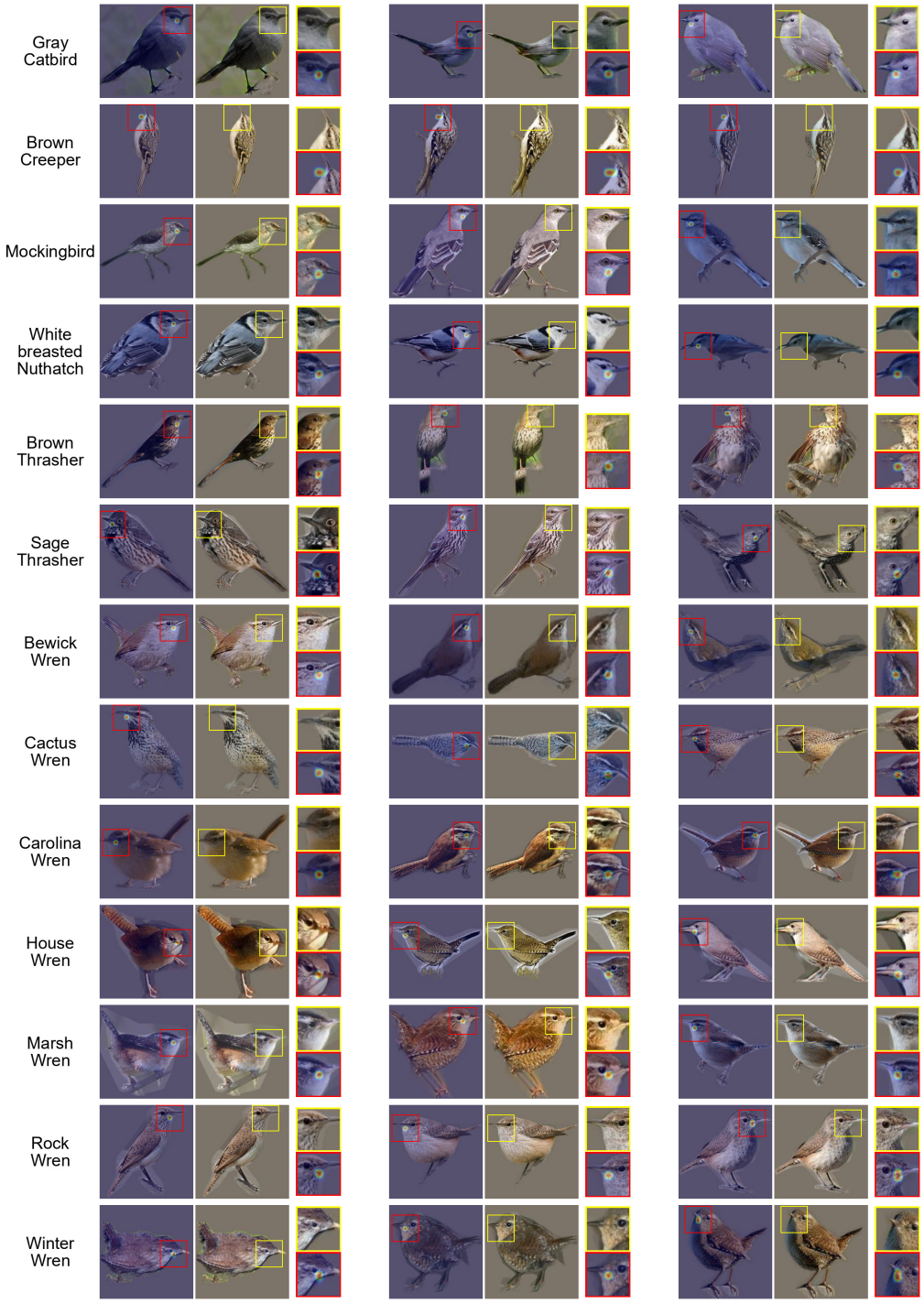


Figure 24: Top-K visualization of a prototype finding commonality between thirteen species of birds sharing a common ancestor. Each row represents the top 3 images from the respective species. For each image, we show the zoomed-in view of the original image as well as the heatmap overlaid image.

Figure 25: Top-K visualization of a prototype finding commonality between five species of birds sharing a common ancestor. Each row represents the top 3 images from the respective species. For each image, we show the zoomed-in view of the original image as well as the heatmap overlaid image.

Figure 26: Top-K visualization of a prototype finding commonality between five species of birds sharing a common ancestor. Each row represents the top 3 images from the respective species. For each image, we show the zoomed-in view of the original image as well as the heatmap overlaid image.

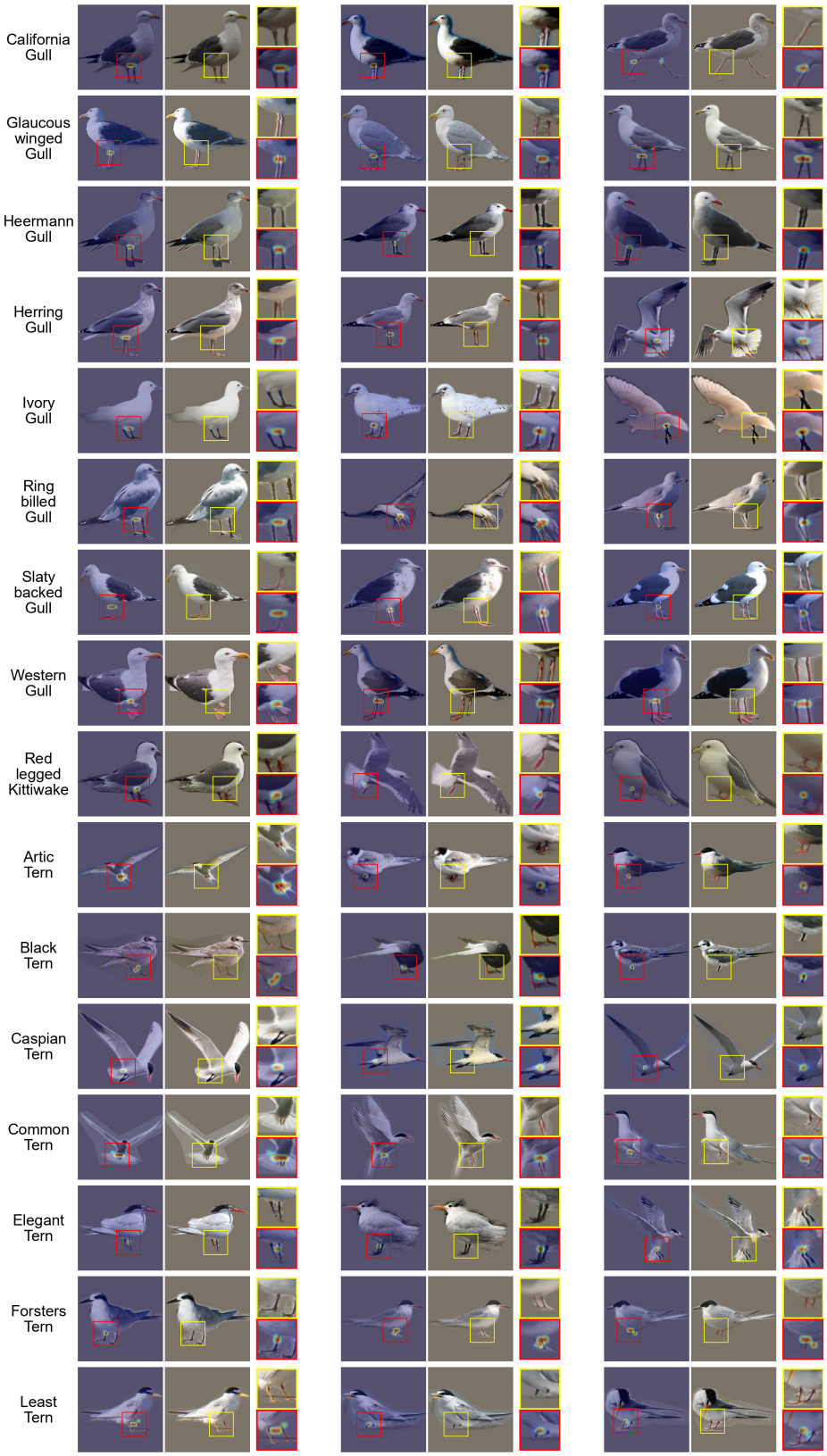


Figure 27: Top-K visualization of a prototype finding commonality between sixteen species of birds sharing a common ancestor. Each row represents the top 3 images from the respective species. For each image, we show the zoomed-in view of the original image as well as the heatmap overlaid image.

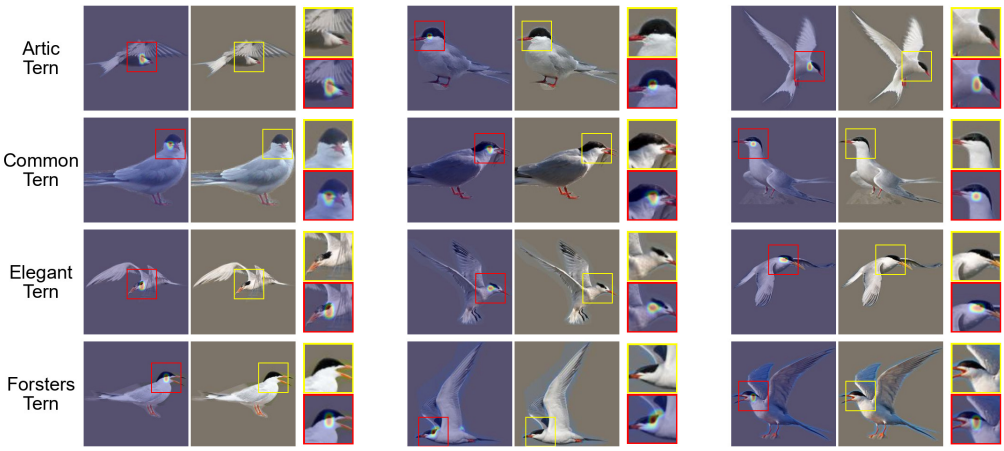


Figure 28: Top-K visualization of a prototype finding commonality between four species of birds sharing a common ancestor. Each row represents the top 3 images from the respective species. For each image, we show the zoomed-in view of the original image as well as the heatmap overlaid image.

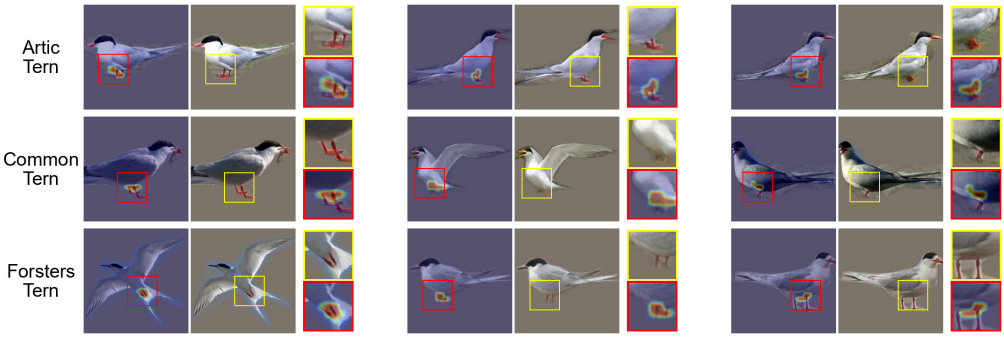


Figure 29: Top-K visualization of a prototype finding commonality between three species of birds sharing a common ancestor. Each row represents the top 3 images from the respective species. For each image, we show the zoomed-in view of the original image as well as the heatmap overlaid image.