ProHOC: Probabilistic Hierarchical Out-of-Distribution Classification via Multi-Depth Networks

Supplementary Material

9. Distributions of hierarchical distances

To analyze the performance of ProHOC in more detail, we compute histograms of hierarchical distances between the ground truth and the predictions. We compute these histograms for ProHOC with EntCompProb as the OOD model. For a more detailed evaluation of the hierarchical distances, we decompose these into an overprediction distance and an underprediction distance as

$$\begin{aligned}\text{dist}_{\mathcal{H}}(y, f(x)) =& \text{dist}_{\mathcal{H}}(\text{LCA}(y, f(x)), f(x)) \\ &+ \text{dist}_{\mathcal{H}}(\text{LCA}(y, f(x)), y)\end{aligned} \tag{24}$$

where y is the ground-truth node, $f(x)$ is the predicted node and $\text{LCA}(y, f(x))$ is the lowest common ancestor of y and $f(x)$, *i.e.*, the deepest node that has both y and $f(x)$ as descendants (where descendants includes itself). We will use $\text{LCA} = \text{LCA}(y, f(x))$ for brevity. With this decomposition, we get the following error cases:

- The prediction is deeper than the LCA: $\text{dist}_{\mathcal{H}}(\text{LCA}, f(x)) > 0$.
- The ground truth is deeper than the LCA: $\text{dist}_{\mathcal{H}}(\text{LCA}, y) > 0$.
- The prediction is a descendant of the ground truth: $\text{dist}_{\mathcal{H}}(\text{LCA}, f(x)) > 0$ and $\text{dist}_{\mathcal{H}}(\text{LCA}, y) = 0$. We denote this case *pure overprediction*.
- The prediction is an ancestor of the ground truth: $\text{dist}_{\mathcal{H}}(\text{LCA}, f(x)) = 0$ and $\text{dist}_{\mathcal{H}}(\text{LCA}, y) > 0$. We denote this case *pure underprediction*.

Figures 3 to 5 illustrates these concepts.

The decomposed hierarchical distances are shown in Figures 6 to 8 where each sample in the respective test sets contributes to a histogram entry.

For OOD data, we observe both over- and underpredictions. Notably, pure overprediction distances of 1 are frequent across all three datasets. In contrast, ID data shows a clear trend of pure underprediction, with many samples being predicted as ancestors to the ground truth. As discussed in Sec. 5.4, ProHOC with EntCompProb generally demonstrates lower ID performance compared to the other models. However, these histograms reveal that the low ID performance is primarily due to predicting ancestors to the ground truth, a behavior that may be acceptable in some applications.

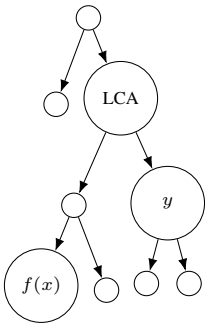

Figure 3. Prediction example: $\text{dist}_{\mathcal{H}}(\text{LCA}, f(x)) = 2$, $\text{dist}_{\mathcal{H}}(\text{LCA}, y) = 1$.

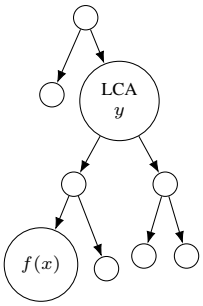

Figure 4. Prediction example: $\text{dist}_{\mathcal{H}}(\text{LCA}, f(x)) = 2$, $\text{dist}_{\mathcal{H}}(\text{LCA}, y) = 0$. This represents a *pure overprediction*.

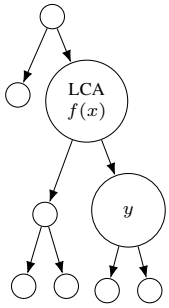

Figure 5. Prediction example: $\text{dist}_{\mathcal{H}}(\text{LCA}, f(x)) = 0$, $\text{dist}_{\mathcal{H}}(\text{LCA}, y) = 1$. This represents a *pure underprediction*.

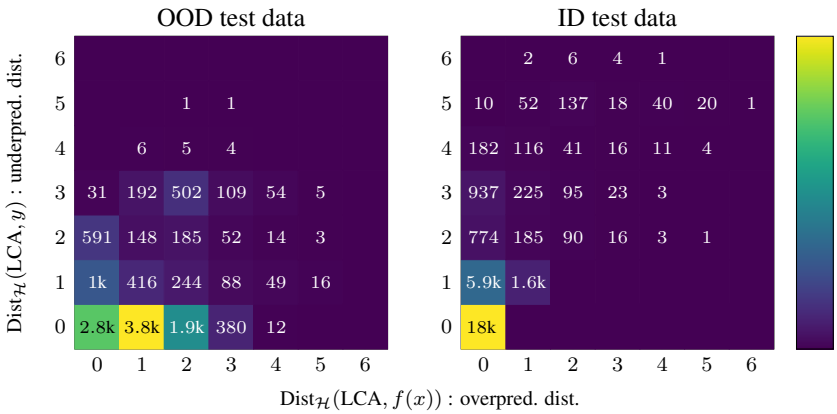

Figure 6. Hierarchical distances: iNaturalist19.

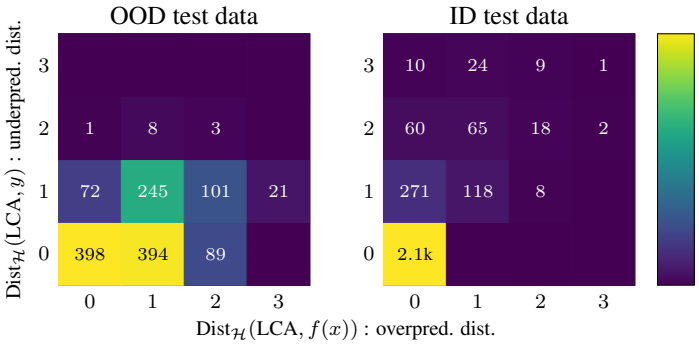

Figure 7. Hierarchical distances: FGVC-Aircraft.

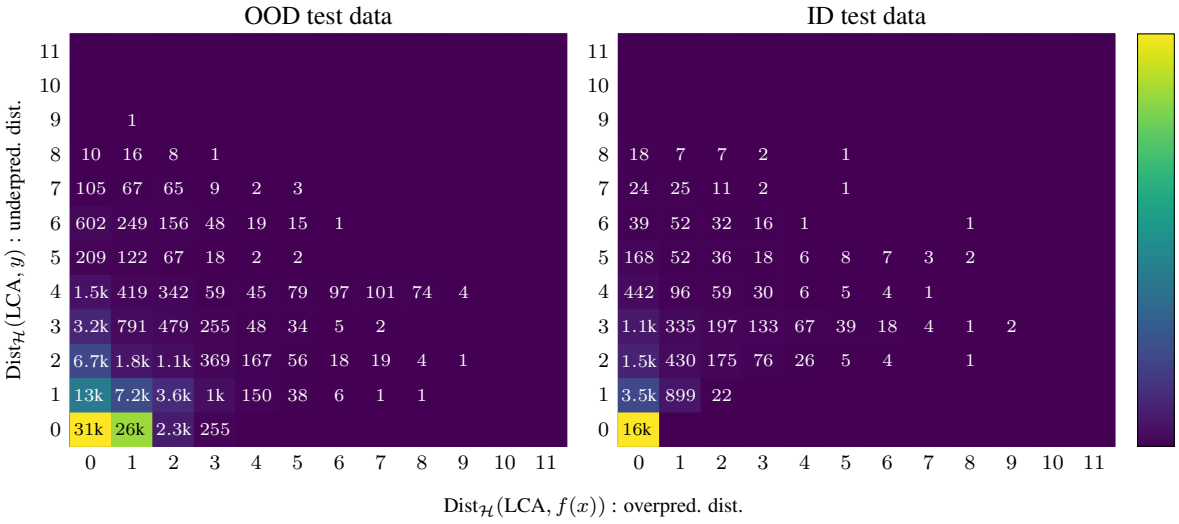

Figure 8. Hierarchical distances: SimpleHierImageNet.

10. Easy and hard OOD classes

For a more qualitative evaluation of the performance of ProHOC, we look at which OOD classes get the best and worst performance. Specifically, Tab. 7 shows the top three and bottom three mean hierarchical distances for OOD classes across each test set. We can see relatively large differences between the easy and hard classes for all datasets, with SimpleHierImageNet displaying the largest spread.

Figures 9 to 14 shows images from the ID and OOD descendants for the top and bottom-performing classes in Tab. 7. Note that these figures do not display the full hierarchy or all the descendants of the particular nodes. For FGVC-Aircraft, Figure 9 shows that the OOD sample of Boeing 737 closely resembles the ID descendants, making it easy to predict correctly. Conversely, for the hard class shown in Figure 10, the ID descendants consist of smaller aircraft, whereas the OOD sample is a large passenger plane with few common visual features to the ID descendants, making it challenging to predict accurately.

For the easy and hard examples of iNaturalist19 shown in Figures 11 and 12 we again see that the ID and OOD descendants in the easy example display strong visual similarities. For the hard example, the flowers differ significantly in color and shape. Additionally, there are many other flower species in the iNaturalist19 dataset, making OOD samples as in Figure 12 challenging.

SimpleHierImageNet has both the easiest and the hardest classes across all our datasets. The OOD samples for Oscine bird (Figure 13) get a low mean hierarchical distance of 0.337. We hypothesize that this class is easy because, as in the easy examples above, its descendants share clear visual features, such as body shape, tail, and beak. However, there are also distinct visual features for distinguishing between the descendants, such as colors and patterns, making it easy to identify a sample as part of the group while distinguishing it from the specific ID descendants.

On the opposite end of the spectrum is the Game equipment class (Figure 14) with a mean hierarchical distance of 4.217. While the model potentially could recognize the round shapes of the balls, the images in these categories tend to be cluttered with various objects and people, making it challenging to identify common features. Additionally, SimpleHierImageNet has, *e.g.*, categories corresponding to clothing that could confuse when there are people in the images.

Table 7. The top and bottom hierarchical distances per class.

OOD Class	Mean $\text{dist}_{\mathcal{H}}(f(x), y)$
iNaturalist19	
Genus: Enallagma	0.43
Genus: Viola	0.45
Genus: Aminata	0.45
Phylum: Angiospermae	2.17
Class: Aves	2.20
Genus: Lysimachia	2.66
FGVC-Aircraft	
Family: Boeing 737	0.53
Manufacturer: Douglas Aircraft Company	0.56
Family: Airbus A320	0.61
Manufacturer: McDonnell Douglas	1.40
Manufacturer: Fokker	1.99
Manufacturer: de Havilland	2.02
SimpleHierImageNet	
Oscine bird	0.34
Insect	0.48
Aquatic bird	0.51
Cat	3.23
Kitchen appliance	3.52
Game equipment	4.22

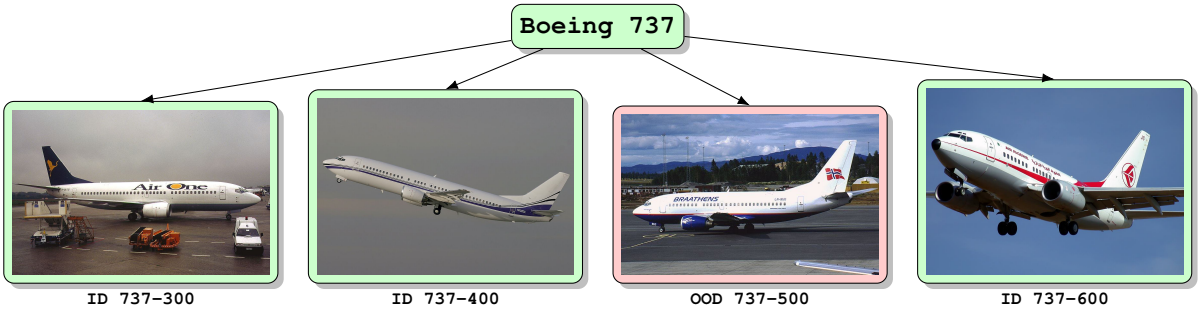

Figure 9. Easy OOD: FGVC-Aircraft.

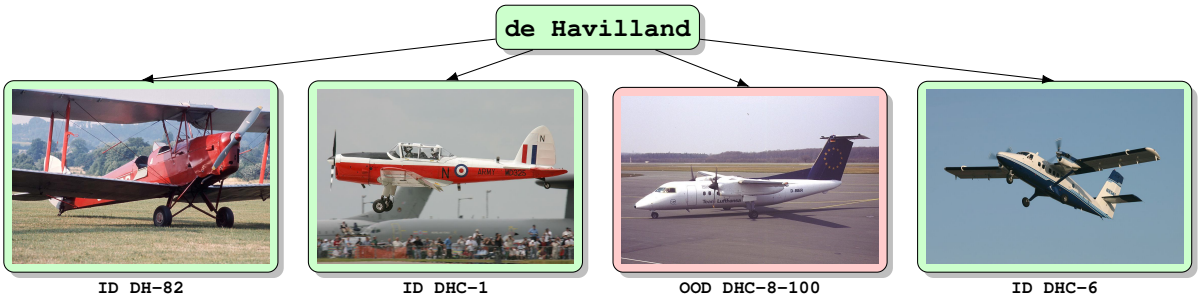

Figure 10. Hard OOD: FGVC-Aircraft.

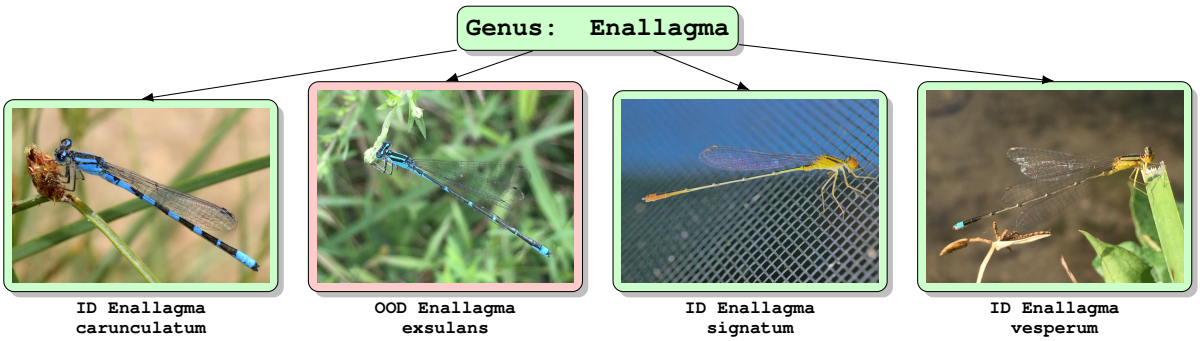

Figure 11. Easy OOD: iNaturalist19.

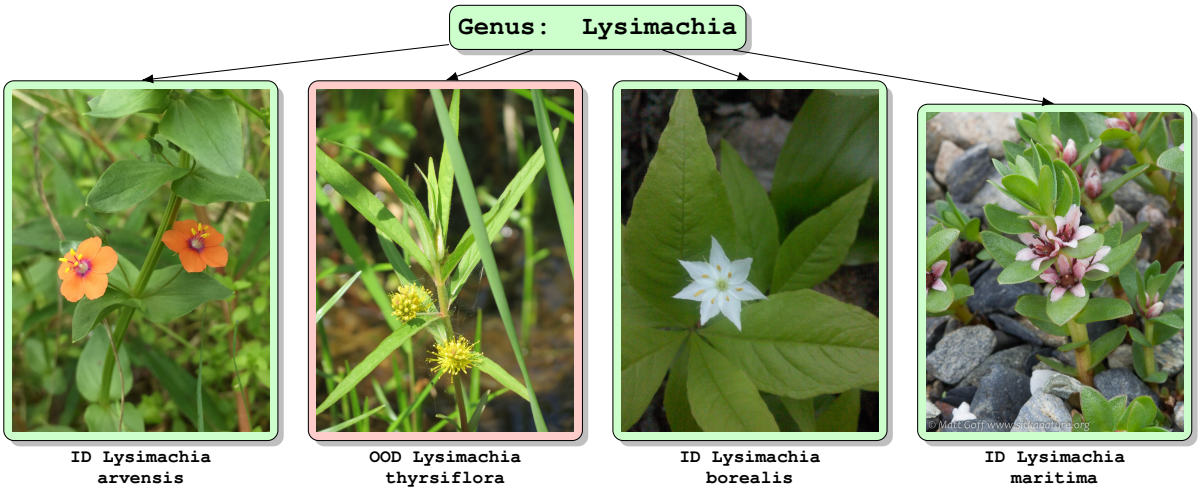

Figure 12. Hard OOD: iNaturalist19.

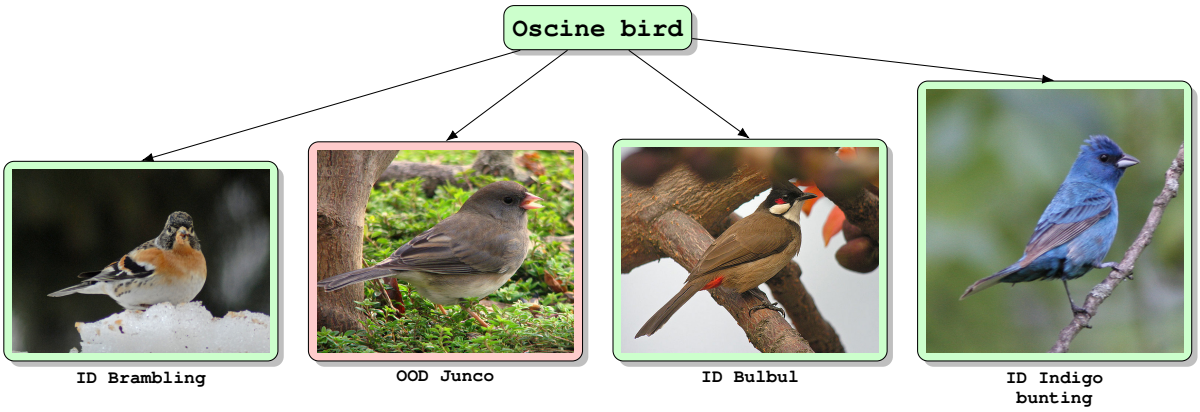

Figure 13. Easy OOD: SimpleHierImageNet.

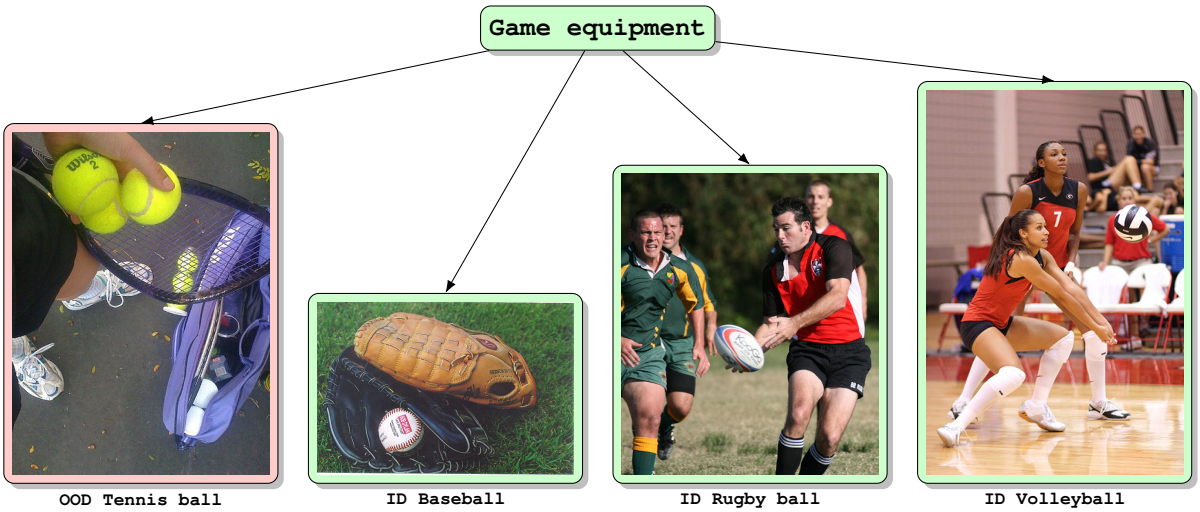

Figure 14. Hard OOD: SimpleHierImageNet.

Table 8. Comparing ProHOC with the ResNet50 backbone and the DINOv2 ViT-L/14 backbone. Results using the ResNet50 backbone are gathered from Tab. 3. Excluding the oracle model, the best results are **boldfaced**.

	Backbone	BAcc_{id} ↑	BAcc_{ood} ↑	MixBAcc ↑	BMHD_{id} ↓	BMHD_{ood} ↓	MixBMHD ↓
SIMPLEHIERIMAGENET							
Depth oracle	ResNet50	79.7	72.5	76.1	0.82	1.05	0.93
Depth oracle	DINOv2 ViT	88.9	81.1	85.0	0.40	0.79	0.60
ProHOC (CompProb)	ResNet50	67.8	19.2	43.5	0.92	1.61	1.27
ProHOC (CompProb)	DINOv2 ViT	85.8	18.6	52.2	0.40	1.50	0.95
ProHOC (EntCompProb)	ResNet50	62.5	30.3	46.4	0.96	1.45	1.21
ProHOC (EntCompProb)	DINOv2 ViT	81.5	34.6	**58.0**	0.42	1.30	**0.86**
iNATURALIST19							
Depth oracle	ResNet50	72.4	75.9	74.2	0.85	0.82	0.83
Depth oracle	DINOv2 ViT	76.8	85.6	81.2	0.58	0.48	0.53
ProHOC (CompProb)	ResNet50	66.1	18.0	42.0	0.77	1.34	1.06
ProHOC (CompProb)	DINOv2 ViT	72.2	23.7	47.9	0.49	1.12	0.81
ProHOC (EntCompProb)	ResNet50	57.7	35.6	46.7	0.78	1.10	0.94
ProHOC (EntCompProb)	DINOv2 ViT	60.1	49.6	**54.9**	0.54	0.82	**0.68**
FGVC-AIRCRAFT							
Depth oracle	ResNet50	84.7	67.6	76.1	0.49	0.67	0.58
Depth oracle	DINOv2 ViT	85.6	61.0	73.3	0.42	0.82	0.62
ProHOC (CompProb)	ResNet50	80.1	17.1	48.6	0.41	1.25	0.83
ProHOC (CompProb)	DINOv2 ViT	67.4	27.0	47.2	0.54	1.16	0.85
ProHOC (EntCompProb)	ResNet50	78.0	22.7	**50.3**	0.41	1.21	0.81
ProHOC (EntCompProb)	DINOv2 ViT	55.6	44.8	50.2	0.63	0.96	**0.80**

11. ProHOC with DINOv2 ViT

All results in the main paper are obtained from the ResNet50 architecture due to its widespread use in image classification research. ProHOC, however, is architecture-agnostic, requiring only that the architecture produces a probability vector over classes, making it compatible with any SOTA architecture. To demonstrate ProHOC's transferability to other architectures and highlight the performance gains from using a stronger image backbone, we conduct experiments with ProHOC using image features from a frozen DINOv2 ViT-L/14 backbone [23]. In this setup, the multi-depth models are replaced with independent MLPs that take DINOv2 features as input. For SimpleHierImageNet and iNaturalist19, we use four-layer MLPs with a hidden dimension of 512 and a batch size of 512. For FGVC-Aircraft, we use single-layer classification heads and a batch size of 128 due to the smaller dataset size. All models are trained for 300 epochs with an initial learning rate of 0.01, decayed to zero at the end of training using a cosine schedule.

The results from training ProHOC with DINOv2 ViT-L/14 are shown in Tab. 8. We see big performance improvements compared to the ResNet50 models on SimpleHierImageNet and iNaturalist19, indicating that ProHOC can leverage the capacity of a stronger backbone model. The EntCompProb model again outperforms CompProb with the DINOv2 backbone. On FGVC-Aircraft, the results from ResNet50 and DINOv2 are closer. Interestingly, the ResNet50 oracle model outperforms DINOv2 for OOD classification, suggesting it captures features relevant for OOD predictions that DINOv2 does not. Nevertheless, the overall performance on FGVC-Aircraft remains similar between ResNet50 and DINOv2.

Note that using a pre-trained backbone like DINOv2 for the hierarchical OOD task changes the preliminaries of the problem. Unlike the ResNet50 models, which encounter OOD data only at test time, the DINOv2 backbone has been pre-trained on all our evaluated datasets (including the OOD classes), albeit without labels. This gives DINOv2 an inherent advantage. Therefore, the key takeaway from these results is not a direct comparison between the ResNet50 and ViT architectures, but that ProHOC can benefit from the stronger data representations provided by DINOv2.

12. ID performance of multi-depth networks

Table 9 shows the ID accuracies of the multi-depth networks used to obtain the results in Tab. 3. Table 9 also shows the number of nodes assigned to each network. As expected, we see a strong correlation between depth and accuracy. Note that the leaf accuracy for iNaturalist19 differs from the value in Tab. 3 as Tab. 9 shows unbalanced accuracies.

Table 9. ID accuracies for the multi-depth networks.

Depth d	# classes at d	Acc
iNaturalist19		
1	3	98.7
2	15	97.6
3	58	93.1
4	239	88.9
5	672	78.9
6	721	75.8
FGVC-Aircraft		
1	30	94.3
2	63	90.3
3	80	84.7
SimpleHierImageNet		
1	2	98.3
2	5	97.8
3	43	95.9
4	54	92.5
5	122	88.2
6	240	85.9
7	402	82.4
8	445	80.7
9	471	80.2
10	512	79.6
11	518	79.7

13. SimpleHierImageNet

As discussed in Sec. 5.1, tieredImageNet in its original form is not well-suited for OOD detection in class hierarchies due to several issues. First, it includes sibling classes that do not share common visual features (*e.g.*, *analog clock* and *digital clock*), as well as visually similar classes that are separated by large hierarchical distances (*e.g.*, *laptop computer* and *computer keyboard*). Additionally, it contains many narrow branches, such as parent nodes with only two children (*e.g.*, *duck*), making it difficult to identify common features associated with the parent.

To summarize the desirable characteristics of a hierarchy suited for hierarchical OOD detection, we consider the following criteria:

- Siblings should share visual features.
- Visually similar classes should be separated by small hierarchical distances.
- Internal nodes should have enough children to enable learning of common visual features.

With these criteria in mind, we have reorganized parts of the tieredImageNet hierarchy to form SimpleHierImageNet, a hierarchy better suited for hierarchical OOD detection. Specifically, we have pruned internal nodes and moved parts of the hierarchy to satisfy the listed criteria. Additionally, a few classes from tieredImageNet are completely omitted because they lack clear visual connections to other classes in the tree, making them difficult to place within the hierarchy while satisfying our requirements. The omitted classes are

- n06359193: website
- n03314780: face powder
- n04192698: shield
- n02840245: binder
- n03657121: lens cap
- n04423845: thimble
- n04507155: umbrella
- n03467068: guillotine
- n03544143: hourglass
- n04355338: sundial.

As a result of this curation, we go from 234 internal nodes in the original tieredImageNet to 66 internal nodes in SimpleHierImageNet. The full specification of SimpleHierImageNet is available at https://github.com/walline/prohoc.

Table 10. The number of samples in the respective datasets.

	# ID train	# ID test	# OOD test
FGVC-Aircraft	5333	2667	1332
SimpleHierImageNet	665877	25900	104452
iNaturalist19	156768	28078	12659

14. Dataset details

In Tab. 10, we specify the number of samples in each dataset. The OOD test set for SimpleHierImageNet is large because it is expanded using the OOD classes from the original ImageNet training split. The OOD subsets used in the experiments are listed in Tabs. 11 to 13 and are also defined at https://github.com/walline/prohoc. These listed classes represent leaf nodes in the original datasets but subsets of these combine to form OOD data associated with higher levels of the tree.

As a last post-processing step, after defining the ID and OOD subsets, we prune the ID hierarchy by removing nodes with only one child. Specifically, we connect the single child directly to the grandparent and remove the intermediate node. The motivation for this pruning is that we consider it unrealistic for the model to learn the difference between a node and its only child.

Table 11. OOD categories for FGVC-Aircraft.

v-737-500
v-737-700
v-747-400
v-767-200
v-767-300
v-767-400
v-A319
v-A330-200
v-A330-300
v-A340-200
v-A340-300
v-A340-500
v-A340-600
v-Challenger_600
v-DC-6
v-DC-9-30
v-DHC-8-100
v-DHC-8-300
v-E-195
v-Fokker_50

Table 12. OOD categories for SimpleHierImageNet as WordNet IDs.

n01534433	n02091635	n02110185	n02883205	n03662601
n02088094	n02091831	n02123394	n03866082	n03673027
n02088238	n02092002	n02397096	n02794156	n02814533
n02088364	n02092339	n02128925	n04548280	n03670208
n02088466	n01855672	n02422106	n03773504	n03345487
n02088632	n02012849	n02481823	n09246464	n04560804
n02089078	n02093991	n02487347	n04515003	n03770679
n02089867	n02017213	n01494475	n02676566	n04604644
n02089973	n02096177	n02643566	n07715103	n02793495
n02090379	n01688243	n02169497	n03394916	n02727426
n02090622	n02098105	n02256656	n07718472	n03089624
n02090721	n01728920	n02279972	n03804744	n02825657
n02091032	n02099429	n07768694	n03642806	n04398044
n02091134	n01744401	n03207941	n02979186	n04285008
n02091244	n02108422	n04542943	n04409515	n04370456
n02091467	n02106166	n03980874	n03179701	n02410509

Table 13. OOD categories for iNaturalist19 with IDs as specified in iNaturalist19.

nat0996	nat0490	nat0400	nat0610	nat0431	nat0239	nat0207	nat0014	nat0055
nat0997	nat0491	nat0881	nat0611	nat0434	nat0240	nat0208	nat0015	nat0056
nat0998	nat0492	nat0882	nat0612	nat0723	nat0241	nat0209	nat0016	nat0057
nat0999	nat0493	nat0883	nat0613	nat0891	nat0242	nat0210	nat0017	nat0058
nat1000	nat0494	nat0884	nat0614	nat0975	nat0243	nat0211	nat0018	nat0059
nat1001	nat0495	nat0885	nat0615	nat0190	nat0244	nat0318	nat0019	nat0060
nat1002	nat0496	nat0886	nat0616	nat0166	nat0245	nat0296	nat0020	nat0061
nat1003	nat0497	nat0887	nat0617	nat0201	nat0246	nat0297	nat0021	nat0062
nat1004	nat0498	nat0888	nat0618	nat0257	nat0247	nat0298	nat0022	nat0063
nat1005	nat0499	nat0889	nat0619	nat0258	nat0248	nat0299	nat0150	nat0064
nat1006	nat0500	nat0890	nat0620	nat0259	nat0249	nat0300	nat0068	nat0065
nat1007	nat0501	nat0866	nat0583	nat0260	nat0250	nat0301	nat0069	nat0066
nat1008	nat0502	nat0867	nat0591	nat0261	nat0251	nat0302	nat0070	nat0067
nat1009	nat0448	nat0836	nat0388	nat0262	nat0252	nat0303	nat0071	nat0032
nat0958	nat0454	nat0565	nat0363	nat0263	nat0253	nat0304	nat0072	nat0038
nat0963	nat0338	nat0567	nat0543	nat0264	nat0254	nat0305	nat0073	nat0000
nat0964	nat0344	nat0621	nat0515	nat0265	nat0255	nat0306	nat0074	nat0004
nat0965	nat0792	nat0622	nat0644	nat0266	nat0256	nat0282	nat0075	
nat0966	nat0776	nat0623	nat0645	nat0212	nat0224	nat0283	nat0076	
nat0967	nat0777	nat0628	nat0646	nat0213	nat0225	nat0284	nat0077	
nat0968	nat0778	nat0596	nat0647	nat0214	nat0226	nat0285	nat0078	
nat0969	nat0779	nat0597	nat0648	nat0215	nat0227	nat0286	nat0079	
nat0970	nat0780	nat0598	nat0649	nat0216	nat0228	nat0287	nat0043	
nat0971	nat0781	nat0599	nat0650	nat0217	nat0229	nat0288	nat0044	
nat0972	nat0782	nat0600	nat0651	nat0218	nat0230	nat0289	nat0045	
nat0917	nat0783	nat0601	nat0652	nat0219	nat0231	nat0290	nat0046	
nat0910	nat0784	nat0602	nat0653	nat0220	nat0232	nat0291	nat0047	
nat0668	nat0785	nat0603	nat0654	nat0221	nat0233	nat0292	nat0048	
nat0669	nat0786	nat0604	nat0655	nat0222	nat0234	nat0293	nat0049	
nat0684	nat0787	nat0605	nat0803	nat0223	nat0202	nat0294	nat0050	
nat0688	nat0788	nat0606	nat0810	nat0235	nat0203	nat0295	nat0051	
nat0469	nat0732	nat0607	nat0818	nat0236	nat0204	nat0315	nat0052	
nat0481	nat0762	nat0608	nat0830	nat0237	nat0205	nat0012	nat0053	
nat0486	nat0765	nat0609	nat0417	nat0238	nat0206	nat0013	nat0054	