Appendix

We provide more information here.

A. Additional Experiments

A.1. Re-weighting and Temperature Scaling

As a top level loss for RAC, we make use of unscaled logit adjustment exclusively, with no reweighting (i.e. $\alpha_y = 1 \quad \forall y \in \mathcal{Y}$) and no temperature scaling ($\tau = 1$) in Eq. (3). This loss is theoretically well-grounded [38], and is appealing due to its simplicity. Nevertheless, other works have noted that the combination of logit adjustment with re-weighting often leads to higher empirical performance [1]. While optimizing the top-level loss is not the focus of RAC, we include here a comparison of RAC performance under various re-weighting schemes when combined with logit adjustment for completeness. Specifically, we consider inverse log

$$\alpha_y = \frac{1}{\log N_y} \tag{6}$$

and inverse square-root

$$\alpha_y = \frac{1}{\sqrt{N_y}} \tag{7}$$

class-frequency based re-weighting of individual sample losses.

We confirm that the same effect is present for RAC on Places365-LT, with overall accuracy increasing with the use of both re-weighting schemes, with improvement most pronounced for tail classes (Table S1). However, this trend does not hold on iNat.

For Places365-LT, we also we perform a sweep across τ, to evaluate if the same effect can be achieved with manual temperature scaling (Fig. S1). Higher τ does result in slightly higher overall accuracy, however this effect is minor in comparison to re-weighting. This divergence from theory is likely due to the non-separability of many classes in Places365-LT due to the high label noise.

Re-weighting	Many	Med	Few	All
Places365-LT				
None	**49.72**	49.34	40.46	47.75
Inverse sqrt.	47.12	**49.58**	**47.65**	**48.32**
Inverse log	44.88	51.35	47.54	48.29
iNat				
None	**75.92**	**80.48**	81.07	**80.24**
Inverse sqrt.	71.13	80.02	81.19	79.56
Inverse log	67.17	80.04	**81.67**	79.35

Table S1. Effect of common re-weighting schemes when combined with logit adjustment for RAC's top-level loss.

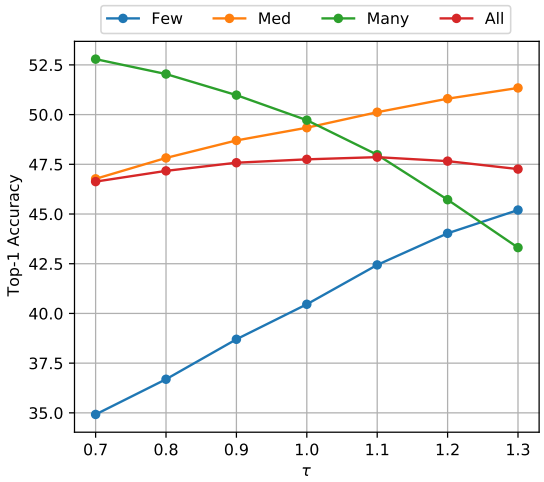

Figure S1. Effect of τ within the LACE loss on balanced performance on the Places365-LT dataset.

A.2. Index Ablations

We examine the effect of the distance metric and index type on RAC's lookup performance and speed in Table S2. To quantify error induced by an approximate index, we include the lookup accuracy on the index content itself (training set) in addition to the validation accuracy. Query Time (QT) includes encoding samples with a ViT-B-16 model, which is the primarily overhead on small indexes.

We observe that the choice of distance metric (ℓ_2 vs. cosine) has little effect, as may be expected, give the high dimentionality of the index. Cosine distance does introduce a minor computational overhead due to the need to normalize the embeddings prior to querying. The drop in accuracy due to use of an approximate (HNSW) instead of exact k-NN is also minor, but comes with a significant ($2\times$) speedup on large index's. Construction time is constant across all indexes at 0.02μs per sample, with the exception of large-index HNSW, which requires 0.05μ per sample. In summary, performance differences are minor when querying small indexes, however as the index size grows, the choice of HNSW becomes critical to ensure lookup time does not bottleneck training.

A.3. Per-class Accuracy on iNat

We include the same per-class visualization presented in the main body for Places365-LT (Fig. 2), for iNat. Note that for iNat only 3 samples are present for each class in the validation set, hence the square-wave appearance of the plots (Fig. S2). Nevertheless, the same trend is clearly visible in the sliding window moving average, with the retrieval

Index Size	Index Type	Distance	Test Many	Med	Few	Top-1	Top-5	Train Top-1	QT (ms/sample)
62.5k	Exact	L_2	39.97	26.74	18.65	29.91	53.81	99.97	1.12
62.5k	Exact	Cosine	39.84	26.51	18.03	29.64	53.31	99.96	1.14
62.5k	HNSW	L_2	39.89	26.51	18.03	29.66	53.32	99.79	1.06
62.5k	HNSW	Cosine	39.57	26.26	17.68	29.37	52.83	99.79	1.07
11.2M	Exact	L_2	-	-	-	-	-	-	7.96
11.2M	HNSW	L_2	-	-	-	-	-	-	3.01

Table S2. Index ablations on Places365-LT. QT indicates Query Time. Large-sample indices are filled with the ImageNet21k dataset.

module performing best on tail classes and the base network largely focusing on the many and mid-frequency classes.

B. Further Details

For **Places365-LT** we use the training and validation splits during development and report final numbers on the test set, with no validation samples used during final training. For **iNaturalist2018**, following prior work, we report results on the released validation split, as labels for the test set are not publicly available.

In the main text, we use several common model variants. While the architectures for these models are standard, we specify the high-level design choices in Tables S4 and S5. These choices are consistent across both the 224×224 and 384×384 variants.

All training is carried out on $8\times$ 32GB A100 GPU's. We largely follow the procedures outlined in [45], with the following alterations. We finetune with AdamW [35] instead of SGD, and make use of low-magnitude RandAugment [6] alone for data augmentation, with no Color-Jitter, Mixup,

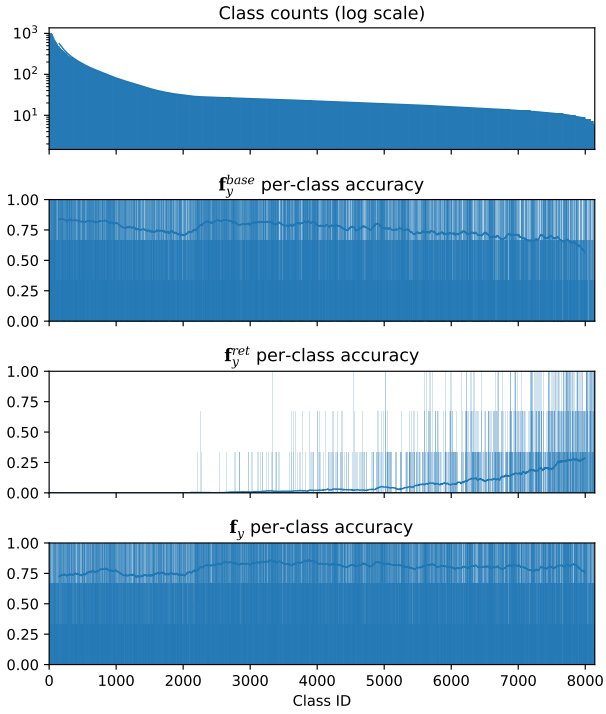

Figure S2. Per-class top-1 accuracy on iNat from each branch's output. The 300 sample moving average over classes (solid line) is shown for clarity.

	Places365-LT	iNat
Samples (train)	62,500	437,513
Samples (test)	36,500	24,426
Classes	365	8,142
Imbalance Factor	500	996

Table S3. Dataset Details

Hyperparameter	ViT-B-16	ViT-B-32
Patch size	16	32
Depth	12	12
Embedding Dimension	768	768
Attention Heads	12	12
Parameters	85.8M	87.4M

Table S4. ViT model architecture details.

Hyperparameter	RN50	RN152d
Input size	$244 \times 224 \times 3$	$256 \times 256 \times 3$
Head	Avg. Pool, FC	Avg. Pool, FC
Convolutions	standard	standard
Stem Convolutions	1 layer, 3×3	3 layer, 3×3
Stem width	32	(128, 128, 128)
Layers	[3, 4, 6, 3]	[3, 8, 36, 3]
Pool size	7×7	8×8
Num. features	2048	2048
Parameters	23.5M	58.2M

Table S5. ResNet model architecture details.

	Places365-LT	iNat
Batch Size	200	50
Learning Rate	5e-5	1e-4
Epochs	30	20

Table S6. Dataset specific training hyperparameters

Hyperparameter	Value
Global normalization means	[0.5, 0.5, 0.5]
Global normalization stds	[0.5, 0.5, 0.5]
Crop (train and test)	0.95
Distributed	DDP 8 GPU's
LR schedule	cosine
Min LR	1e-7
Warmup LR	1e-7
Warmup epochs	5
Optimizer	adamw
Beta 1	0.9
Beta 2	0.999
Eps.	1e-8
Gradient clipping	L_2 Norm
Gradient clipping magnitude	1.0
RandAugment magnitude	1
RandAugment layers	3
RandAugment noise std.	0.5
Weight decay	0.02
Label smoothing	0.1
Stochastic depth	0.1
Random erase prob.	0.0
Color jitter	0.0
Random scale	[0.75, 1.33]
Random crop	No
Horizontal flip prob.	0.5
Random rotation	No
Mixed precision level	O2

Table S7. Training hyperparameters for the ViT models

Cutmix, RandomErase or Augmix applied. Full hyperparameters are shown in Table S7. While we found the use of Mixup and Cutmix does boost performance on standard ViTs trained under a BalCE loss, their use of combined targets requires special treatment to make compatible with the LACE loss, which requires hard targets in order to assign the class adjustment. While one approach may be to apply the 'merged' class adjustment, the performance benefit is marginal and hence we simply did not include either approach in RAC's data augmentation pipeline.

C. Retrieval Branch Visualization

While Table 4 quantifies top-1 performance of the retrieval branch, non-exact match snippets will also effect RAC as they are still likely to be informative. If 'plane', 'runway', 'concrete', 'sky', 'propeller' etc. are returned for example, it is not difficult for $\mathbf{B}$ to place a high score on 'airport'. We visualize the returned strings for random samples by distance and frequency in Figs. S3 and S4. For all runs, $k = 30$ as in the main work.

Each column displays (from left to right):

1. query image x_q,
2. retrieved labels sorted by distance (distance shown in brackets, exact matches colored green),
3. retrieved labels and occurrence counts, sorted from most to least frequent,
4. histogram of distances to all returned samples,
5. the correct label.

We restrict the lists of retrieved labels to the top eight for visualization purposes. Note that as the distance metric is cosine, a higher value corresponds to more similar samples.

D. Label Overlap with Pretraining Datasets

Places365-LT is commonly used to evaluate long-tail learning, however almost all approaches fine-tune ImageNet pretrained encoders, and hence there is the potential for label overlap. As far as we are aware this has not been specifically examined or noted in the literature. ImageNet1K in fact shares 26 exact-match labels with Places365-LT, (many of which are in the tail such as 'valley', 'viaduct' etc.) and many close matches. Futhermore, the average $L2$ distance between roBERTa-L encodings of the Places365-LT labels and the closest ImageNet1k label is 0.816. For context, the distance between 'dog' $\leftrightarrow$ 'greyhound' is 0.911 and 'dog' $\leftrightarrow$ 'toolkit' is 1.303.

This indicates there is significant semantic overlap between ImageNet1k and the Places365-LT dataset and it is more an evaluation of long-tail transfer, where the challenge is to associate already learned representations with the correct labels extremely quickly, rather than learn new classes from scratch. iNat, on the other hand, does evaluate long-tail learning more reliably by providing a dataset where there is almost no overlap due to its highly fine-grained nature. RAC performs very well on both and especially well on the long-tail of iNat.

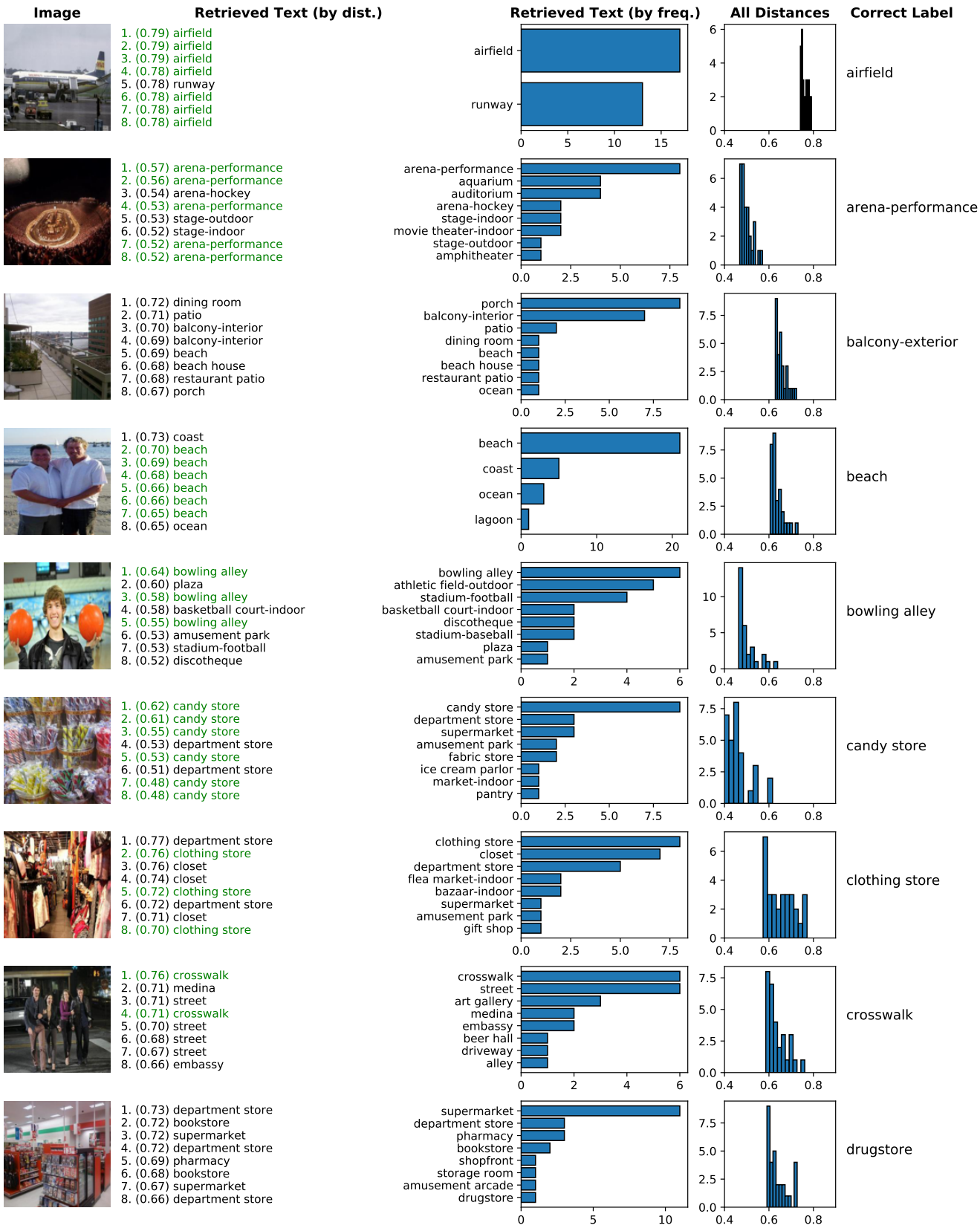

Figure S3. Retrieval branch visualization for randomly drawn samples from Places365-LT. For detailed explanation see Appendix C.

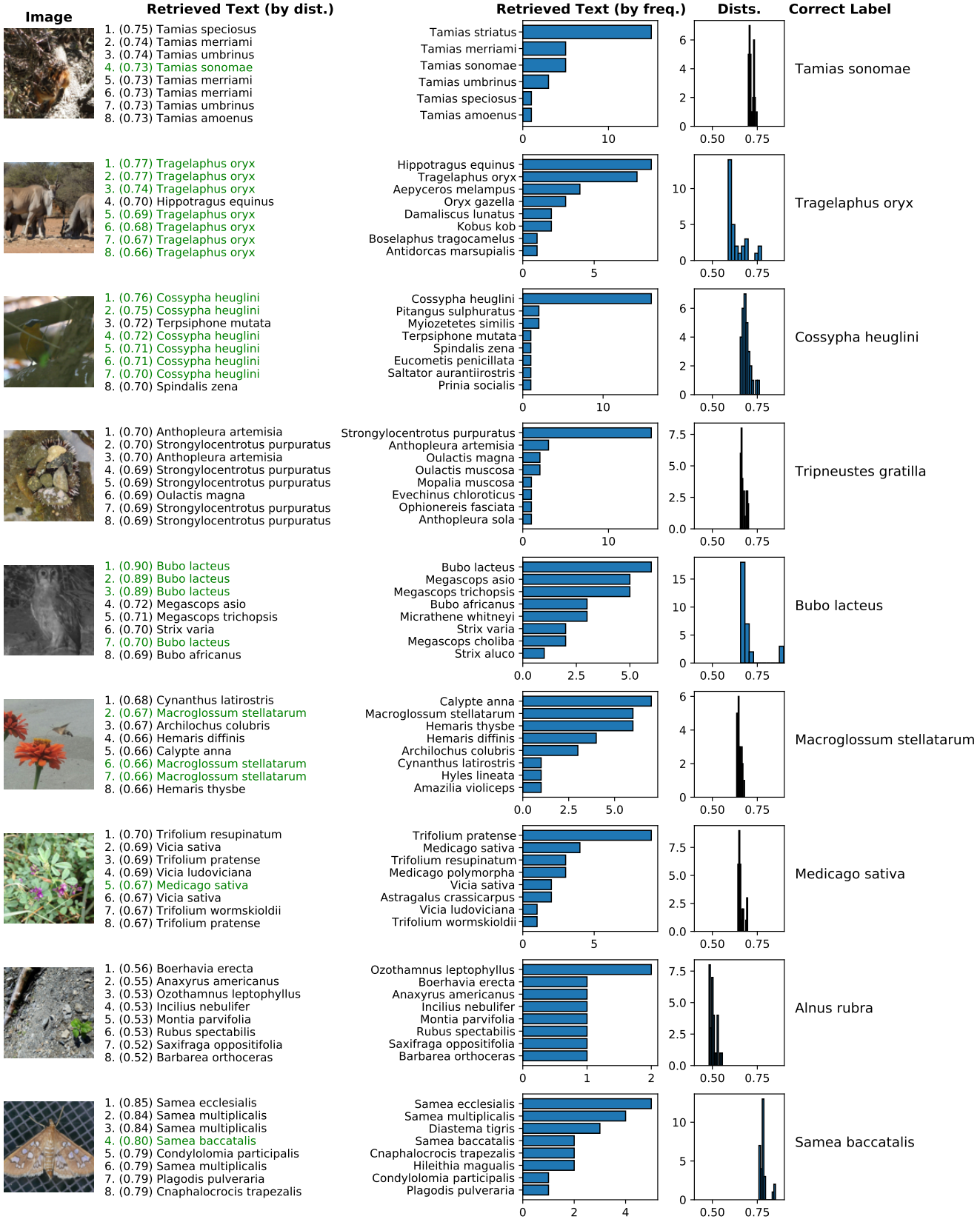

Figure S4. Retrieval branch visualization for randomly drawn samples from iNaturalist2018. For detailed explanation see Appendix C.