On Label Granularity and Object Localization Supplementary Material

Elijah Cole[1,†] Kimberly Wilber[2] Grant Van Horn[3] Xuan Yang[2]
Marco Fornoni[2] Pietro Perona[1] Serge Belongie[4]
Andrew Howard[2] Oisin Mac Aodha[5]

[1]Caltech [2]Google [3]Cornell University
[4]University of Copenhagen [5]University of Edinburgh

[†]ecole@caltech.edu

A Additional Experiments

A.1 Does coarse training help on other datasets?

In the main paper we show that training on coarser labels significantly improves WSOL performance on iNatLoc500. It is reasonable to ask whether we can replicate this pattern on other datasets. In Fig. A1 we present results on FGVC-Aircraft [5], CUB [10], and ImageNet [4]. We give dataset-specific details below.

FGVC-Aircraft. The FGVC-Aircraft dataset consists of images of different kinds of aircraft, which are organized into a label hierarchy with the following tiers, ordered from coarsest to finest: manufacturer, family, and variant. Fig. A1(top) shows that training with coarser labels improves WSOL performance (+4.3 `MaxBoxAccV2`). This shows that the benefits of coarse training are not limited to natural world datasets.
Training details: We use the best hyperparameters for CUB from [1], except that we train for 10 epochs and decay the learning rate every 3 epochs.

CUB. Fig. A1(middle) shows that training with coarser labels improves WSOL performance (+4.4 `MaxBoxAccV2`). This indicates that our observations on iNatLoc500 in the main paper generalize to images collected under different protocols i.e. iNaturalist user photos vs. iconic images crawled from Flickr. Unlike iNatLoc500, we do not see a drop in performance at the coarsest level. This is is consistent with our previous findings because CUB contains only birds, so its hierarchy terminates before reaching the level of granularity where iNatLoc500 performance drops.
Training details: We use the filtered version of CUB as described in Sec. C and train with the best hyperparameters for CUB from [1].

ImageNet. Fig. A1(bottom) shows no benefit to coarsening the labels for ImageNet. This is consistent with our claim in the main paper that the ImageNet hierarchy does not measure granularity, and motivates the development of better label hierarchies for datasets like ImageNet in future work.
Training details: Unlike iNatLoc500, FGVC-Aircraft, and CUB, the ImageNet label hierarchy has leaf nodes at many different depths. To accommodate this, we must modify the coarsening procedure from the main paper. Instead of coarsening every leaf node at every step, we only coarsen leaf nodes which are at the deepest level of the hierarchy. Each granularity level is named `cX` where `X` $\in \{0, 1, 2, \ldots\}$ is the number of times this coarsening has been applied. To speed up training we sample 200 images per category for D_w. We use the filtered version of ImageNet as described in Sec. C and train with the best hyperparameters for ImageNet from [1].

A.2 What is the impact of longer training schedules?

In [1], the authors design their WSOL training schedules so that CUB and OpenImages30k use the same computational budget. They use a budget of 300k images processed, which equates to 50 epochs for CUB and 10 epochs for OpenImages30k. To respect this budget, in the main paper we train on iNatLoc500 for 2 epochs (276k images processed). In Fig. A2 we see that a longer training schedule can improve performance slightly (Family, Phylum) or significantly (Species, Genus, Order, Class). However, the pattern is the same whether we train for 2 epochs or 10 epochs, i.e. performance drops for labels that are too coarse or too fine.

A.3 How does WSOL performance depend on the IoU threshold?

Throughout the main paper we use the `MaxBoxAccV2` metric proposed by [1]. This metric averages performance over three IoU thresholds: 30%, 50%, and 70%. In Fig. A3 we show the performance of CAM on iNatLoc500 separately for each IoU threshold. Not surprisingly, we see that performance decreases significantly as the IoU threshold becomes more demanding (i.e. larger). We also observe that, regardless of the IoU threshold, the best performance is obtained at a label granularity that is neither too fine nor too coarse. In the right panel of Fig. A3 we see that the relative performance improvement is larger for more demanding IoU thresholds. This may be because there is less room to improve for "easier" IoU thresholds.

A.4 How stable are the CAM results?

Each result in the main paper is the result of a single run, so it is important to quantify how much test performance varies when we re-train. In Fig. A4 we show the results of re-training CAM on iNatLoc500 five times at each granularity level with identical hyperparameters. The standard deviations at different granularity

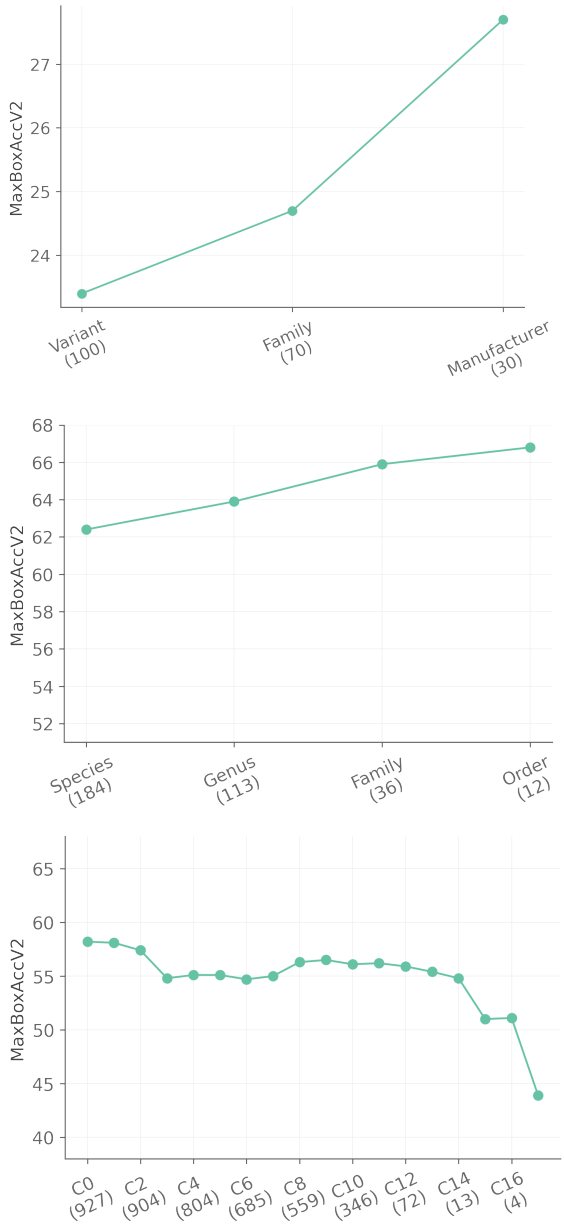

Fig. A1. WSOL performance as a function of training set label granularity for FGVC-Aircraft (top), CUB (middle) and ImageNet (bottom). Like iNatLoc500, FGVC-Aircraft and CUB show significant gains at coarser granularities. There is no apparent benefit for ImageNet, which lacks a consistent label hierachy.

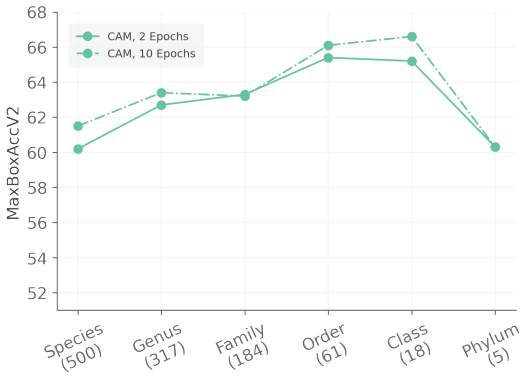

Fig. A2. Comparison of our standard training schedule (2 epochs, reducing learning rate after 1 epoch) and a longer training schedule (10 epochs, reducing learning rate every 3 epochs) for CAM on iNatLoc500. Training for longer does not change the observation that coarser labels result in better localization.

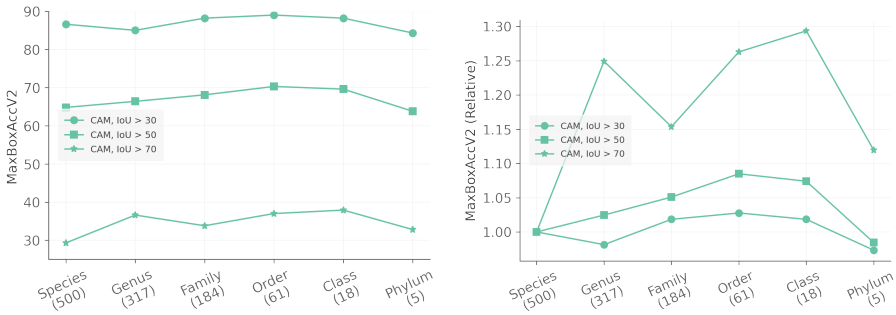

Fig. A3. CAM performance on iNatLoc500 as a function of label granularity and IoU threshold. The left panel shows absolute performance. The right panel shows performance relative to the species-level performance, which is the traditional baseline approach. More specifically, the right panel is generated by normalizing each curve in the left panel by its left-most endpoint.

levels range from ~ 0.2 to ~ 0.8, which is much smaller than the effect sizes we discuss in the main paper. Interestingly, training seems to be most stable for the best-performing coarse-grained levels (order and class), and least stable for the genus level.

A.5 What is the effect of additional hyperparameter tuning?

In their paper, [1] searches over 30 random hyperparameter sets for each WSOL method. We use a less computationally intensive protocol. For iNatLoc500, we start from their best hyperparameters for ImageNet and re-optimize the learn-

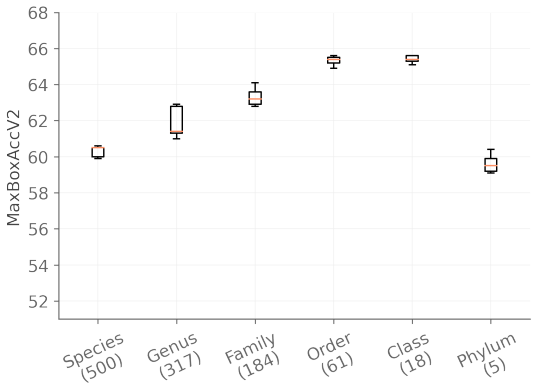

Fig. A4. Distribution of CAM performance at each granularity level of iNatLoc500 for five runs with identical hyperparameters. The orange line denotes the mean.

ing rate by searching over $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. To quantify the performance difference between our reduced hyperparameter search and the full search in [1], we conduct both procedures on iNatLoc500 using CAM at the species level. The full hyperparameter search (30 hyperparameter sets) achieves 60.8 `MaxBoxAccV2` compared to 60.2 `MaxBoxAccV2` for our abbreviated hyperparameter search (5 hyperparameter sets). As expected, the additional hyperparameter optimization provides an improvement for CAM but the difference is surprisingly modest. We would expect a similar boost to occur for any granularity level. We also note that the gap may be greater for methods with more hyperparameters to tune. We provide the learning rates used in our paper in Table A3.

B Dataset Construction Details for iNatLoc500

In this section we detail the process of merging and cleaning data from iNat17 and iNat21 to produce iNatLoc500.

Species matching. In total there are 4486 species names that occur in both iNat17 and iNat21. We discard any images which do not correspond to a species shared by both datasets. We also omit any species that does not have bounding box annotations. In particular, this means that we discard all plant species, since iNat17 does not have any bounding boxes for plants.

Removing duplicate observations. Each image on the iNaturalist platform has an associated `observation_id` which corresponds to a unique encounter with an individual plant or animal. We find all observation IDs which occur in both iNat17 and iNat21 and we remove all of the corresponding images from iNat21. It is important to remove duplicates at the `observation_id` level instead of the image level, since an iNaturalist observation may be associated with multiple similar but distinct images of the same individual organism.

Instance count filtering. Since our focus is object localization (as opposed to detection), any images with multiple bounding box annotations are discarded.

Box size filtering. Any image whose box is smaller than 32 pixels in either dimension is removed. In addition, any image whose box width (height) is more than 96% of the image width (height) is removed. This step speeds up the annotation process by filtering out a significant number of "bad" images. Very small boxes are problematic because annotators are more likely to make mistakes, while very large boxes tend to be extreme close-ups.

Split considerations. While the majority of observations on iNaturalist are associated with only one image, some do have multiple images. When splitting the fully supervised images into D_f and D_{test} we ensure that all of the images for one observation go into exactly one split. This is important because images from the same observation can be highly similar.

Manual Annotation. Well-annotated validation and test sets are essential for reliable model selection and benchmarking. The image-level fine-grained class labels reflect the consensus of the iNaturalist community, and like prior iNaturalist datasets [9,8] we assume they are correct. However, the bounding box annotations were crowd-sourced with non-expert workers. We therefore manually validate the bounding box annotations for the images in the D_f and D_{test} splits. Images with any of the issues listed below were excluded from the dataset. Note that the distribution of images in D_f and D_{test} is likely to be somewhat different than the distribution of images in D_w due to this cleaning process. Examples of problematic images are given in the supplementary material.

- *Missing instances.* Images with multiple bounding box annotations are filtered out before annotation cleaning. Unfortunately, the bounding box annotations for an image are sometimes incomplete, which means that an image with one bounding box annotation for a species can contain multiple instances of that species. Images with multiple instances of the labeled species are removed.

- *Inaccurate bounding boxes.* Some bounding boxes are too large or too small, e.g. boxes which miss appendages such as legs or tails or boxes which only contain the face of the animal. Images with inaccurate bounding boxes are removed. We also remove any images for which it is unclear whether or not the bounding box is correct, which may occur when an image is blurry or poorly illuminated.

- *Indirect evidence.* iNaturalist accepts images showing *indirect evidence* of an animal (e.g. footprints, feathers, droppings), not just images of the animal itself. We omit images which show only indirect evidence of an animal. We also omit images of animal carcasses, which are not uncommon for e.g. deer.

- *Body part close-ups.* Some images in iNaturalist are clearly intended to show the structure of some specific body part in scientific detail, such as an image of a paw next to a ruler. We omit these images.

C Label Hierarchies

We visualize the label hierarchies for CUB, ImageNet, and iNatLoc500 in Fig. A5. Producing our final hierarchies for CUB and ImageNet required some care. We give details below.

CUB. CUB was not released with a label hierarchy, so we constructed one. We start by attempting to map each category to a node on the tree of life, like iNatLoc500. CUB consists of 200 bird categories, where some of these categories correspond to species (e.g. `Black-footed Albatross`) and some do not (e.g. the genus `Sayornis` or umbrella terms like `frigatebird`). We discard any CUB category whose name could not be unambiguously mapped to a single species. By checking these species names against the iNaturalist taxonomy, we obtained the genus, family, order, and class for each species. All bird species belong to the class `Aves`, so this is the root node of the label hierarchy. Since we retained only species-level categories, every leaf node lies at the same distance from the root node. Our CUB label hierarchy has 184 leaf nodes.

ImageNet. ImageNet is equipped with a label hierarchy based on WordNet [6]. One problem with this hierarchy is that some nodes have multiple parents, which violates the assumptions of the label coarsening procedure outlined in the main paper. We remedy this using a simple greedy approach in which we iterate over the nodes with multiple parents in some fixed (but arbitrary) order and delete all but one parent node. In particular, for each node with multiple parents we perform the following operations:

1. Choose a parent and compute the number of leaf nodes that are still reachable if that parent is retained and the others are deleted. Repeat for each parent.
2. Keep the parent node for which the greatest number of leaf nodes remain reachable from the root.
3. Delete the other parent nodes.
4. Delete any descendants of deleted nodes which are no longer reachable from the root.

After executing this process, we obtain a label hierarchy in which each (non-root) node has a unique parent. Our ImageNet label hierarchy has 927 leaf nodes.

D Descriptive Statistics

D.1 Class Imbalance and Label Granularity

We give basic statistics on the distribution of images over categories for iNatLoc500 at different granularity levels in Table A1. We also visualize the distribution of images over categories at different granularity levels in Fig. A7. At the species level, the categories are approximately balanced, but the spread between the largest and the smallest category is much larger for coarser label sets.

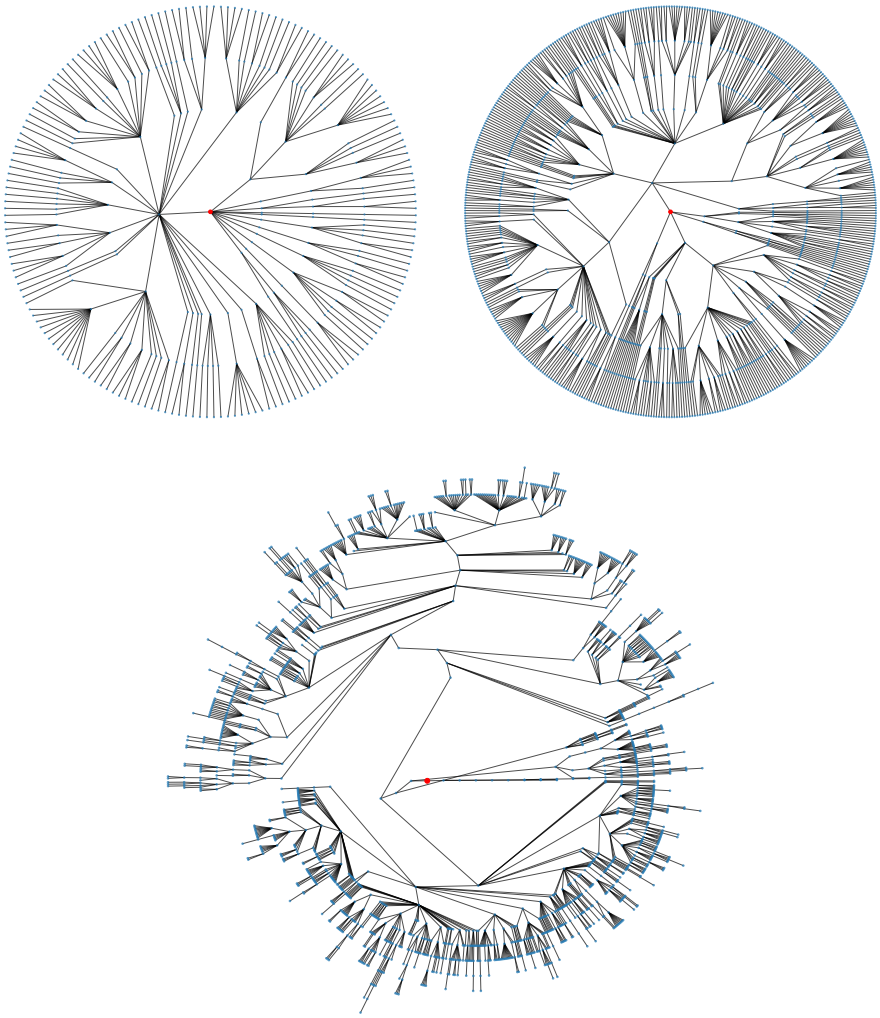

Fig. A5. Label hierarchies for CUB (top left), iNatLoc500 (top right), and ImageNet (bottom). Root nodes are shown in red. The hierarchies for CUB and iNatLoc500 have uniform depths (4 and 6, respectively). The hierarchy for ImageNet is considerably more irregular.

D.2 Box Size

In Fig. A6 we compare the box size distributions for iNatLoc500, ImageNet, and CUB. For each curve in Fig. A6 we compute the area of each box, divide the box areas by the corresponding image sizes, and compute the CDF. The box distribution for iNatLoc500 seems to interpolate between the the box distributions

Table A1. Summary statistics for iNatLoc500 at each granularity level. For each granularity level, we provide the number of categories as well as the minimum, maximum, and mean number of images per category. We also calculate the imbalance factor, which is the size of the largest class divided by the size of the smallest class [3]. Refer to Fig. A7 for a visualization of the distribution of images over categories at each granularity level.

Granularity	# Categories	Min	Max	Mean	Imbalance
Species	500	149	307	276	2.1
Genus	317	149	3575	435	24.0
Family	184	149	7113	750	47.7
Order	61	149	23947	2262	160.7
Class	18	265	29741	7666	112.2
Phylum	5	1345	93576	27599	69.6

for CUB and ImageNet, e.g. iNatLoc500 has more "small" boxes than CUB but not as many as ImageNet.

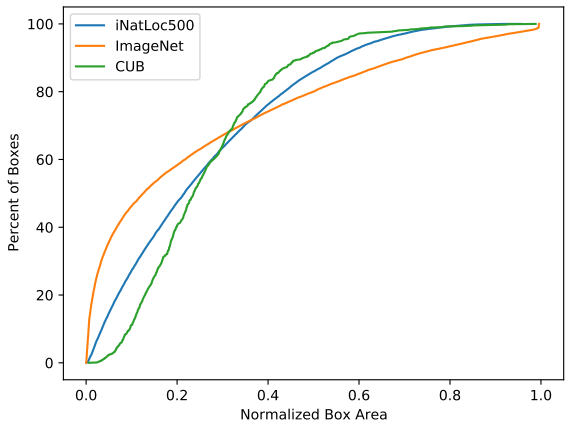

Fig. A6. Comparison of CDFs of box sizes for iNatLoc500, ImageNet, and CUB. All box sizes are normalized by the size of the image.

E Performance Scores

For ease of comparison we provide the raw `MaxBoxAccV2` scores for each WSOL method (and CAM-Agg) at each granularity level in Table A2.

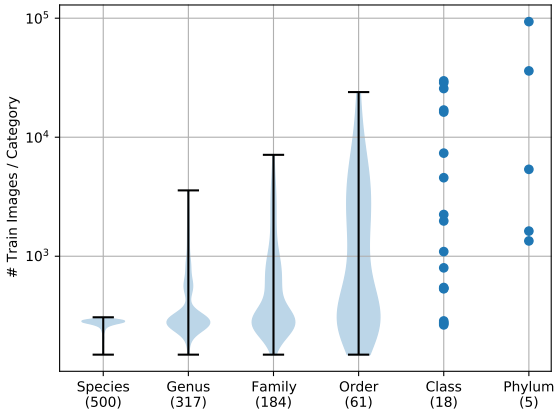

Fig. A7. Distribution of images over categories at different granularity levels for iNat-Loc500. We show violin plots for Species, Genus, Family, and Order. Class and Phylum contain only a small number of categories so we can show each point individually. See Table A1 for summary statistics at each granularity level.

F Implementation Details

F.1 Qualitative Analysis of WSOL

In this section we define the terms used in the qualitative analysis figures in the main paper. In what follows we choose the threshold t to be the optimal threshold for an IoU of 0.50, as defined by [1].

- `Area of Predicted Box`: The area of the predicted box divided by the area of the ground truth box. The predicted box is computed using a threshold t.
- `GT Box Activation`: The sum of the heatmap pixels inside the ground truth box divided by the sum of the heatmap pixels outside the ground truth box.
- `Number of Connected Components`: The number of connected components in the predicted heatmap after it has been binarized with threshold t.

F.2 WSOL Methods

We consider six standard WSOL methods in this work: CAM [14], HaS [7], ACoL [12], SPG [13], ADL [2], and CutMix [11]. We leave the details of those methods to their respective papers. For each WSOL method, we use the training procedures and optimal hyperparameters used by [1] for ImageNet. The only exceptions are as follows. First, we always use enlarged 28×28 feature maps, instead of letting the choice between 14×14 and 28×28 be an additional hyperparameter. Second, we use a weight decay of 10^{-5} instead of 10^{-4}. Third, we always re-optimize the learning rate by searching over the set

Table A2. `MaxBoxAccV2` scores for different WSOL methods trained at different levels of granularity. As in the main paper, ACoL is excluded due to poor performance. We also include provide scores for CAM-Agg, an alternative method of using using granularity information for WSOL described in the main paper.

Method	Species	Genus	Family	Order	Class	Phylum
CAM	60.2	62.7	63.3	65.4	65.2	60.3
HaS	60.0	61.5	63.4	64.1	62.3	52.1
ACoL	-	-	-	-	-	-
SPG	60.7	63.5	63.2	64.6	62.4	55.6
ADL	58.9	63.4	63.7	63.9	64.3	59.8
CutMix	60.1	63.3	63.7	66.7	64.1	61.1
CAM-Agg	60.2	59.8	58.6	55.0	48.8	40.1

$\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ and using the value of `MaxBoxAccV2` on D_f to select the best one. The best learning rate values for each WSOL method and granularity level are provided in Table A3 and the method-specific hyperparameters can be found in Table A4. We summarize the rest of the training details, which match [1], below.

Architecture. All methods use an ImageNet-pretrained ResNet-50 backbone with an input resolution of 224×224.

Image preprocessing. Training images are resized to 256×256, randomly cropped to 224×224, then horizontally flipped with probability 0.5. At test time, images are simply resized to 224×224. All images are normalized according to ImageNet statistics.

Optimization. We train using SGD with Nesterov momentum, a momentum parameter of 0.9, and a batch size of 32. The learning rate for the final linear classifier layer is set to be $10\times$ larger than the learning rate for the rest of the network. For fairness, [1] trains on each dataset for a number of epochs which equates to processing 300k images. To respect this criterion, we train iNatLoc500 for 2 epochs (276k images processed) and decay the learning rate by a factor of 10 after the first epoch.

Evaluation. The search space for the optimal heatmap threshold consists of 1000 linearly spaced values between 0 and 1. Note that all heatmaps are min-maxed normalized before evaluation, so their values fall in $[0, 1]$. Final `MaxBoxAccV2` numbers are an average over three IoU thresholds: 30, 50, and 70.

F.3 Center Baseline

We perform baseline experiments using the "center" baseline for WSOL introduced in [1], which simply generates a centered Gaussian heatmap for each image. Since [1] did not fully specify the implementation of their center baseline, our re-implementation may differ slightly. We opt for a simple implementation which does not depend on the image shape. Specifically, we generate an image

Table A3. Best learning rates for WSOL methods at different granularity levels. Learning rates are selected from $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ based on the value of `MaxBoxAccV2` on `train-fullsup`. Note that learning rates for ACoL at coarser granularity levels are omitted due to poor performance.

Method	Species	Genus	Family	Order	Class	Phylum
CAM	10^{-2}	10^{-1}	10^{-2}	10^{-2}	10^{-2}	10^{-2}
HaS	10^{-2}	10^{-1}	10^{-2}	10^{-2}	10^{-2}	10^{-2}
ACoL	10^{-3}	-	-	-	-	-
SPG	10^{-1}	10^{-1}	10^{-1}	10^{-1}	10^{-1}	10^{-2}
ADL	10^{-1}	10^{-1}	10^{-1}	10^{-2}	10^{-2}	10^{-2}
CutMix	10^{-2}	10^{-1}	10^{-2}	10^{-2}	10^{-2}	10^{-2}

Table A4. Method-specific hyperparameters used for iNatLoc500. These are the same hyperparameters used by [1] for ImageNet.

Method	Hyperparameters
CAM	N/A
HaS	`drop_rate` $= 0.09$, `drop_area` $= 31$
ACoL	`erasing_threshold` $= 0.79$
SPG	$\delta_l^{B1} = 0.02$, $\delta_h^{B1} = 0.03$, $\delta_l^{B2} = 0.05$, $\delta_h^{B2} = 0.47$, $\delta_l^{C} = 0.29$, $\delta_h^{C} = 0.36$
ADL	`drop_rate` $= 0.68$, `erasing_threshold` $= 0.93$
CutMix	`size_prior` $= 0.10$, `mix_rate` $= 0.93$

$C \in \mathbb{R}^{M \times M}$ where

$$C_{i,j} = \exp\left(-\frac{((i - \frac{M-1}{2})^2 + (j - \frac{M-1}{2})^2)}{2\sigma^2}\right)$$

for the pixel in row i and column j. We then apply min-max normalization to C. We set $M = 224$ and $\sigma = M/4$.

Note that in the continuous domain, the value of σ would not matter, since, for any σ, a square centered box of any size could be obtained by choosing the right heatmap threshold. In practice, the heatmap threshold is optimized over a fixed grid of values. In this case, each value of σ yields a different collection of centered boxes, which results in different performance numbers.

F.4 FSL-Seg: Few-Shot Localization via Segmentation

The FSL-Seg baseline for WSOL was introduced in [1], but they did not fully specify the implementation details so our approach may differ. Our training protocols are identical to those we use for WSOL methods, except for the modifications described below.

Architecture. Like the WSOL methods, we begin with an ImageNet-pretrained ResNet-50 with an input resolution of 224×224. We modify the network by replacing the final fully connected layer with a 1×1 convolution layer with a sigmoid activation. Since the feature maps have shape $2048 \times 28 \times 28$, the output of this modified ResNet-50 is a single "score map" $S \in [0, 1]^{28 \times 28}$.

Loss. We train using a weighted per-pixel binary cross-entropy loss given by

$$\sum_{ij} \left[\frac{Y_{ij}}{\|Y\|_0} \log S_{ij} + \frac{(1 - Y_{ij})}{\|1 - Y\|_0} \log(1 - S_{ij}) \right]$$

where $Y \in \{0, 1\}^{28 \times 28}$ is a binary label mask and $\|Y\|_0$ denotes the number of nonzero values in Y. This weighting has the effect of equally balancing positive and negative labels. For OpenImages30k, binary label masks are directly available. However, CUB, ImageNet, and iNatLoc500 only have bounding box annotations. For these three datasets we compute Y by converting the bounding box annotations into binary masks. Note that these masks are noisy because most objects do not completely fill their bounding boxes.
Optimization. For each dataset we train for 10 epochs and decay the learning rate by a factor of 10 every 3 epochs.

F.5 FSL-Det: Few-Shot Localization via Detection

For FSL-Det we use a Faster-RCNN object detection architecture. We use an off-the-shelf TensorFlow Object Detection API training configuration file originally meant for training a Faster-RCNN model on COCO. Other than changing the input image size and the dataset, we do not modify the architecture or any training procedures. The configuration can be found here:

`https://github.com/tensorflow/models/blob/65407126c5adc216d606d360429fe12ed3c3f187/research/object_detection/configs/tf2/faster_rcnn_resnet50_v1_640x640_coco17_tpu-8.config`

G Manual Annotation

We performed extensive filtering and quality control to produce `train-fullsup` and `test` splits data for iNatLoc500. We show randomly selected examples from iNatLoc500 in Fig. A8. We also show examples of images which were rejected and give the reason in each case in Fig. A9.

H Qualitative Examples

We show some hand-picked predictions for CAM-based WSOL in Fig. A10.

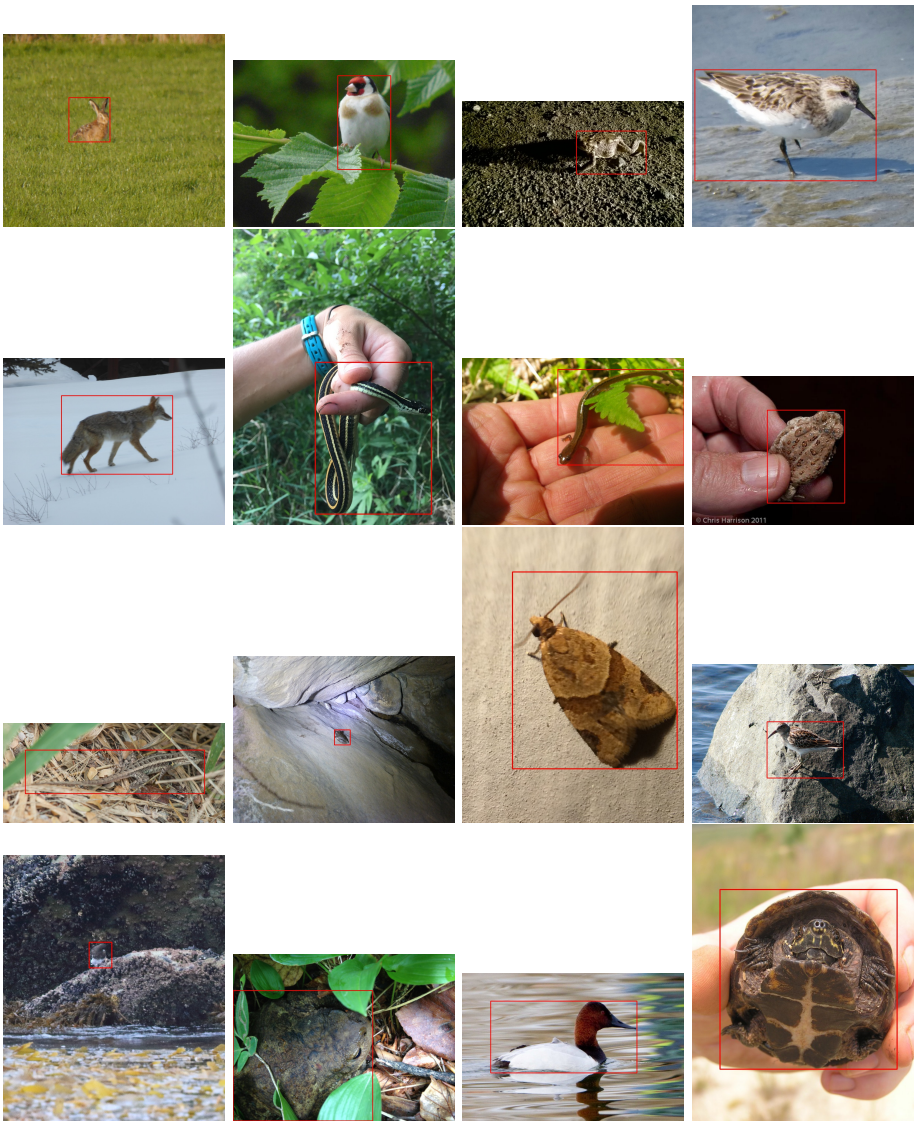

Fig. A8. Randomly selected sample images from the iNatLoc500 dataset.

Fig. A9. Examples of problematic images from iNat17 which were filtered out of the iNatLoc500 dataset. Each image is identified with a tuple (i, j) where i is the row and j is the column. We now describe the problem in each image. (1, 1): The box is too small. (1, 2): The box is too large. (1, 3): The target class is the crab, not the otter, so the box is too large. (1, 4): The box is too large and there are multiple instances of the target species. (2, 1): The box is too large and there are multiple instances of the target species. . (2, 2): The image is an extreme close-up. (2, 3): The correct box is ambiguous due to blurring. (2, 4): The box is too large.

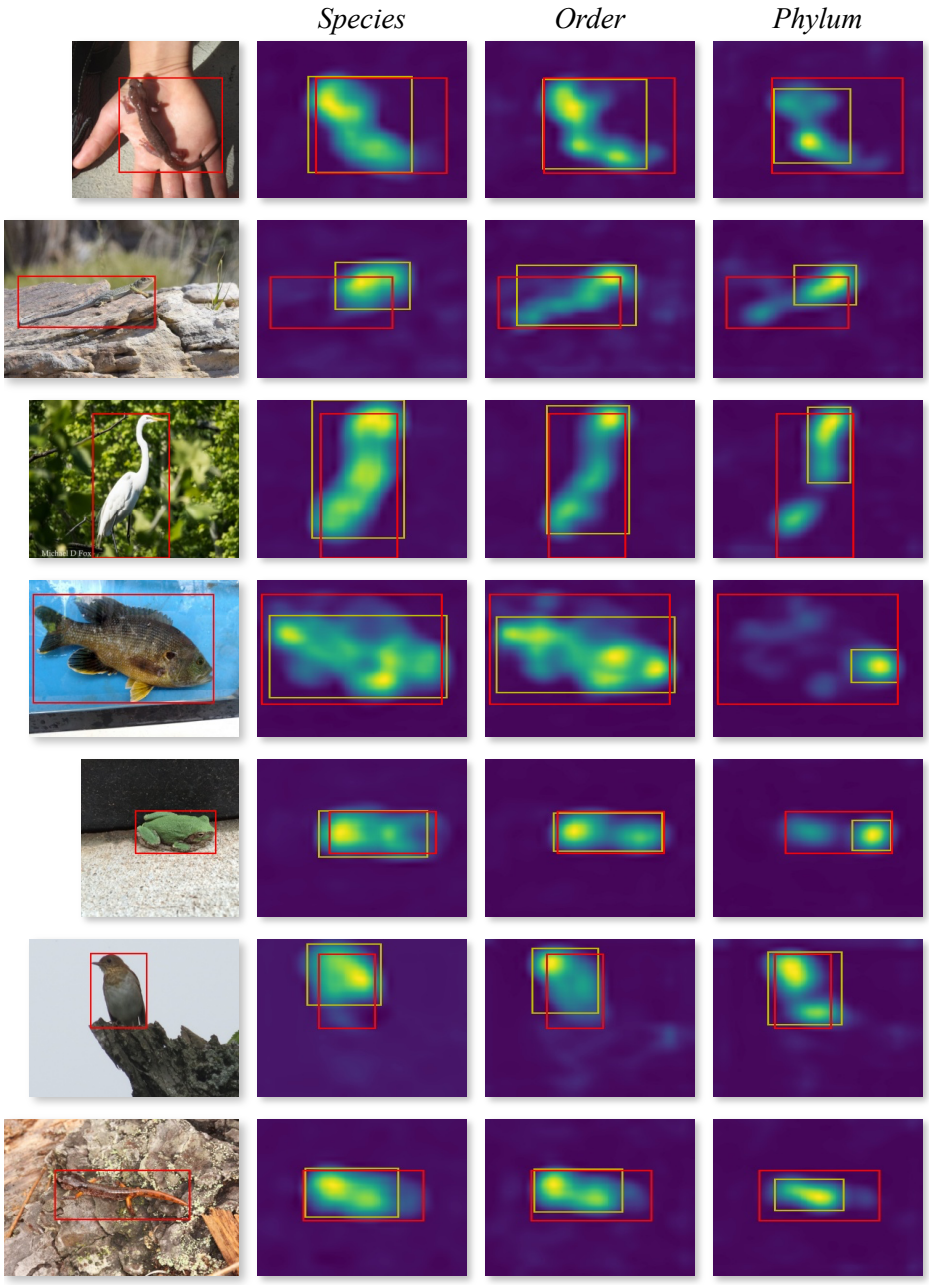

Fig. A10. Examples of CAM-based WSOL predictions at different levels of granularity. In each row we provide activation map for classifiers trained at the phylum, order, and species level. Each activation map shows the ground truth bounding box (red) and WSOL-based bounding box (yellow).

References

1. Choe, J., Oh, S.J., Lee, S., Chun, S., Akata, Z., Shim, H.: Evaluating weakly supervised object localization methods right. In: CVPR (2020)
2. Choe, J., Shim, H.: Attention-based dropout layer for weakly supervised object localization. In: CVPR (2019)
3. Cui, Y., Jia, M., Lin, T.Y., Song, Y., Belongie, S.: Class-balanced loss based on effective number of samples. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9268–9277 (2019)
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)
5. Maji, S., Kannala, J., Rahtu, E., Blaschko, M., Vedaldi, A.: Fine-grained visual classification of aircraft. Tech. rep. (2013)
6. Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM (1995)
7. Singh, K.K., Lee, Y.J.: Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In: ICCV (2017)
8. Van Horn, G., Cole, E., Beery, S., Wilber, K., Belongie, S., Mac Aodha, O.: Benchmarking representation learning for natural world image collections. In: CVPR (2021)
9. Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., Belongie, S.: The inaturalist species classification and detection dataset. In: CVPR (2018)
10. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset. Tech. Rep. CNS-TR-2011-001, California Institute of Technology (2011)
11. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: ICCV (2019)
12. Zhang, X., Wei, Y., Feng, J., Yang, Y., Huang, T.S.: Adversarial complementary learning for weakly supervised object localization. In: CVPR (2018)
13. Zhang, X., Wei, Y., Kang, G., Yang, Y., Huang, T.: Self-produced guidance for weakly-supervised object localization. In: ECCV (2018)
14. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: CVPR (2016)