Keep CALM and Improve Visual Feature Attribution
– Supplementary Material –

Supplementary Materials

Supplementary Materials contain supporting claims, derivations for formulae, and auxiliary experimental results for the materials in the main paper. The sections are composed sequentially with respect to the contents of the main paper.

A. Equivalence of CAM formulations

In §3 of the main paper, we have argued that the formulation in Equation 1 copied below,

$$p(y|x) = \text{softmax}\left(\frac{1}{HW}\sum_{hw} f_{yhw}(x)\right) \tag{a}$$

is equivalent to CNNs with an additional linear layer $W \in \mathbb{R}^{C\times L}$ after the global average pooling (e.g. ResNet):

$$p(y|x) = \text{softmax}\left(\sum_{l} W_{yl}\left(\frac{1}{HW}\sum_{hw}\overline{f}_{lhw}(x)\right)\right) \tag{b}$$

where $\overline{f}$ is a fully-convolutional network with output dimensionality $\overline{f}(x) \in \mathbb{R}^{L\times H\times C}$. This follows from the distributive property of sums and multiplications:

$$\sum_{l} W_{yl}\left(\frac{1}{HW}\sum_{hw}\overline{f}_{lhw}(x)\right) = \frac{1}{HW}\sum_{hw}\sum_{l} W_{yl}\overline{f}_{lhw}(x) \tag{c}$$

where we may re-define $f_{yhw} := \sum_l W_{yl}\overline{f}_{lhw}$ as another fully-convolutional network with a convolutional layer with 1×1 kernels (W_{yl}) at the end.

For networks of the form in Equation b, the original CAM algorithm computes the attribution map by first taking the sum

$$f_{hw} = \sum_{l} W_{\widehat{y}l}\overline{f}_{lhw}(x) \tag{d}$$

and normalizing f as in Equation 2 in main paper:

$$s = \begin{cases} (f^{\widehat{y}}_{\max})^{-1}\max(0, f^{\widehat{y}}) & \text{max [10]} \\ (f^{\widehat{y}}_{\max} - f^{\widehat{y}}_{\min})^{-1}(f^{\widehat{y}} - f^{\widehat{y}}_{\min}) & \text{min-max [6]} \end{cases} \tag{e}$$

Hence, for both training and interpretation, the family of architectures described by Equation 1 subsumes the family originally considered in CAM [10] (Equation b).

B. Derivation of CALM_{EM} objective

In §4.1.2, we have introduced an expectation-maximization (EM) learning framework for our latent variable model. We derive the EM objective in Equation 6 here. Our aim is to minimize the negative log-likelihood $-\log p_\theta(y|x)$. We upper bound the objective as follows.

$$-\log p_\theta(y|x) = -\log\sum_z p_\theta(y,z|x) \tag{f}$$

$$= -\log\sum_z p_{\theta'}(z|x,y)\frac{p_\theta(y,z|x)}{p_{\theta'}(z|x,y)} \tag{g}$$

$$\le -\sum_z p_{\theta'}(z|x,y)\log\frac{p_\theta(y,z|x)}{p_{\theta'}(z|x,y)} \tag{h}$$

$$\le -\sum_z p_{\theta'}(z|x,y)\log p_\theta(y,z|x)\,. \tag{i}$$

The inequalities leading to Equation h and i follow from the Jensen's inequality and the positivity of the entropy, respectively.

We parametrize each term with a neural network. $p_\theta(y,z|x)$ is computed via $g_{yz}\cdot h_z$ and $p_{\theta'}(z|x,y)$ is first decomposed as

$$p_{\theta'}(z|x,y) = \frac{p_{\theta'}(y,z|x)}{\sum_l p_{\theta'}(y,l|x)} \tag{j}$$

and computed with neural networks

$$p_{\theta'}(z|x,y) = \frac{g'_{yz}\cdot h'_z}{\sum_l g'_{yl}\cdot h'_l} \tag{k}$$

where g' and h' are neural networks parametrized with θ'.

C. Training details for CALM

We provide miscellaneous training details for CALM. See §5.1 in the main paper for major training details.

Architecture. We use the ResNet50 as the feature extractor. We enlarge the attribution map size to 28×28 by changing the stride of the last two residual blocks from 2 to 1. Two CNN branches g and h are followed by the feature extractor. The branch g is the one convolutional layer of kernel

size 1 and stride 1 with the number of output channel to be the number of classes. The branch h is composed of one convolutional layer of kernel size 1 and stride 1 with the number of output channel to be 1, followed by the ReLU activation function.

Optimization hyperparameters. We use the stochastic gradient descent with the momentum 0.9 and weight decay 1×10^{-4}. We set the learning rate as (3×10^{-5}, 5×10^{-5}, 7×10^{-5}) for (CUB, OpenImages, ImageNet).

D. More qualitative results

See qualitative results in Figure A for the comparison of attributions against the ground-truth attributions using the counterfactual maps $s^{\text{A}} - s^{\text{B}}$. As in Figure 4 of the main paper, we observe that CALM attains the best similarity with the ground-truth masks. CALM are also the most human-understandable among the considered methods.

We add more qualitative results of the attribution maps s^{A} for the ground-truth class for CUB (Figure B), OpenImages (Figure C), and ImageNet (Figure D). For each case, we do not have the *GT attribution maps* as in Figure A, but we show the *GT foreground* object bounding boxes, or masks if available (OpenImages). Note that the attribution maps (CALM_{EM} and CALM_{ML}) are not designed to highlight the full object extent, while the aggregated versions (+$\mathcal{Y}$) are designed so. We observe that in all three datasets, CALM_{EM} +$\mathcal{Y}$ and CALM_{ML} +$\mathcal{Y}$ tend to generate high-quality foreground masks for the object of interest; for OpenImages, note that "+$\mathcal{Y}$" does not change CALM_{EM} and CALM_{ML} as the supersets $\mathcal{Y}$ are all singletons.

E. Evaluation protocol for WSOL

In §5.4 of the main paper, we have presented experimental results on WSOL benchmarks. In this section, we present metrics, datasets, and validation protocols for the WSOL experiments. We follow the recently proposed WSOL evaluation protocol [1].

Metrics. After computing the attribution s, WSOL methods binarize the attribution by a threshold τ to generate a foreground mask $\mathbf{M} = \mathbb{1}[s_{ij} \geq \tau]$. When there are mask annotations in the dataset, we compute pixel-wise precision and recall at the threshold τ. The `PxAP` is the area under the precision and recall curve at all possible thresholds $\tau \in [0, 1]$. On the other hand, when only the box annotations are available, we generate a bounding box that tightly bound each connected component on the mask $\mathbf{M}$. Then, we calculate intersection over union (IoU) between all pairs of ground truth boxes and predicted boxes at all thresholds $\tau \in [0, 1]$. When there is at least one pair of (ground truth, prediction, τ) with IoU $\geq \delta$, we regard the localization prediction of the attribution as correct. The `MaxBoxAccV2` is the average of three ratios of correct images in the dataset at three $\delta = 0.3, 0.5, 0.7$.

Evaluation protocol. Every dataset in our WSOL experiments consists of three disjoint splits: `train`, `val`, and `test`. The `val` and `test` splits contain images with localization and class labels, while the `train` split contains images only with class labels. We use the `train` split to train the classifier, and tune our hyperparameter by checking the localization performance on the `val` split. Specifically, we only tune the learning rate by randomly sampling 30 learning rates from the log-uniform distribution LogUniform$[5^{-5}, 5^{-1}]$. Then, based on the performance on `val`, we select the optimal learning rate. Finally, we measure the final localization performance on `test` split with the selected learning rate. Since previous WSOL methods in [1] are also evaluated under this evaluation protocol, we can compare fairly our method with the previous methods.

F. Superset $\mathcal{Y}$ for ImageNet classes

In the main paper §5.4, we have discussed the disjoint goals pursued by two tasks, visual attribution and WSOL. The former aims to locate cues that make the class distinguished from the others and typically locates a small part of the object; the latter aims to locate the foreground pixels for objects. To bridge the two, we have considered an aggregation strategy to produce a foreground mask from attribution maps. Our assumption is that attribution maps for different classes for the same family sharing the identical object structure will cover various object parts in the foreground mask. For example, 200 bird classes in CUB share the part structure of "head-beak-breast-wing-belly-leg-tail", but each class will highlight different parts. Combining the attribution maps for the 200 classes will roughly cover all the object parts, providing a higher-quality foreground mask.

For 1000 classes in ImageNet1K, we manually annotate the corresponding supersets $\mathcal{Y}$. We first build a tree of concepts over the classes using the WordNet hierarchy [5]. The leaf nodes correspond to 1000 classes, while the root node corresponds to the concept "entity" that includes all 1000 classes. For each leaf node (ImageNet class) y, we have annotated the superset $\mathcal{Y}$ by choosing an appropriate parent node of y. The parent selection criteria is as follows:

- Choose the parent node $\text{PA}(y)$ of y as close to the root as possible;
- Such that $\mathcal{Y}$, the set of all children of $\text{PA}(y)$, consists of classes with the same object structure as y.

Algorithm. We efficiently annotate the parent nodes by traversing the tree in a breadth-first-search (BFS) manner from the root node, "entity". We start from the direct children (depth$= 1$) of the root node. For each node of depth 1,

we mark if the concept contains classes of the same object structure (`hom` or `het`). For example, the family of classes under the `organism` parent is not yet specific enough to contain classes of homogeneous object structures, so we mark `het`. We continue traversing in depths 2, 3, and so on. If a node at depth d is marked `hom`, we treat all of its children to have the same superset $\mathcal{Y}$ and do not traverse its descendants for depth$>$ d. For example, the `canine` superset of 116 ImageNet classes is reached by following the genealogy of `organism` $\rightarrow$ `animal` $\rightarrow$ `chordate` $\rightarrow$ `mammal` $\rightarrow$ `placental` $\rightarrow$ `carnivore` $\rightarrow$ `canine`. We note that the task is fairly well-defined for humans.

Results. We find 450 supersets in ImageNet1K. 120 of them are non-singleton, consisting of at least two ImageNet classes. The rest 330 classes are singleton supersets. See Table A for the list.

G. A complete version of WSOL results

See Table 5 for the complete version of Table 3 of the main paper. We show the architecture-wise performances for CALM, CAM, and six previous WSOL methods (HaS [4], ACoL [8], SPG [9], ADL [2], CutMix [7], InCA [3]).

In the ImageNet1K dataset, the proposed method achieves competitive performance (62.8%) with CAM (62.4%) and InCA (63.1%). The superset $\mathcal{Y}$ aggregation improves the localization performances for CALM_{EM} (62.5% $\rightarrow$ 62.8%), but it decreases the CAM performances significantly (62.4% $\rightarrow$ 60.6%). For CUB, we observe that CALM_{EM} does not localize the birds effectively without the superset $\mathcal{Y}$ (52.5%). With the superset aggregation, CALM_{EM} attains 65.4%. This is expected behavior because CALM_{EM} attributions often highlight small discriminative object parts (*e.g.* Figure 4 in the main paper). For OpenImages, CALM_{EM} achieves the state-of-the-art performances on all three backbone architectures (61.3%, 64.4%, 62.5%). Since the modified OpenImages [1] consists of classes of unique object structures, the supersets are all singletons. The performances are thus identical with or without the aggregation $+\mathcal{Y}$. In summary, CALM_{EM} on ImageNet is competitive, compared to the state of the art, and is the new state of the art on CUB and OpenImages.

H. More qualitative examples for WSOL

In Figure 6 of the main paper, we have shown the aggregation of attribution maps at different depths of hierarchy for the "canine" class. We show additional qualitative examples of the superset aggregation for other classes in Figure E. We note that the optimal depths that precisely cover the object extents differ across classes.

References

[1] Junsuk Choe, Seong Joon Oh, Seungho Lee, Sanghyuk Chun, Zeynep Akata, and Hyunjung Shim. Evaluating weakly supervised object localization methods right. In *CVPR*, 2020. 2, 3

[2] Junsuk Choe and Hyunjung Shim. Attention-based dropout layer for weakly supervised object localization. In *CVPR*, 2019. 3, 4

[3] Minsong Ki, Youngjung Uh, Wonyoung Lee, and Hyeran Byun. In-sample contrastive learning and consistent attention for weakly supervised object localization. In *ACCV*, 2020. 3, 4

[4] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *ICCV*, 2017. 3, 4

[5] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 1995. 2

[6] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017. 1

[7] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019. 3, 4

[8] Xiaolin Zhang, Yunchao Wei, Jiashi Feng, Yi Yang, and Thomas S Huang. Adversarial complementary learning for weakly supervised object localization. In *CVPR*, 2018. 3, 4

[9] Xiaolin Zhang, Yunchao Wei, Guoliang Kang, Yi Yang, and Thomas Huang. Self-produced guidance for weakly-supervised object localization. In *ECCV*, 2018. 3, 4

[10] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016. 1, 4

Class name	WordNet ID	# Classes	Class name	WordNet ID	# Classes	Class name	WordNet ID	# Classes
canine	n02083346	116	wheel	n04574999	4	power tool	n03997484	2
bird	n01503061	52	swine	n02395003	3	farm machine	n03322940	2
reptile	n01661091	36	lagomorph	n02323449	3	slot machine	n04243941	2
insect	n02159955	27	marsupial	n01874434	3	free-reed instrument	n03393324	2
primate	n02469914	20	coelenterate	n01909422	3	piano	n03928116	2
ungulate	n02370806	17	echinoderm	n02316707	3	lock	n03682487	2
aquatic vertebrate	n01473806	16	person	n00007846	3	breathing device	n02895606	2
building	n02913152	12	firearm	n03343853	3	heater	n03508101	2
car	n02958343	10	clock	n03046257	3	locomotive	n03684823	2
bovid	n02401031	9	portable computer	n03985232	3	bicycle,	n02834778	2
arachnid	n01769347	9	brass	n02891788	3	railcar	n02959942	2
ball	n02778669	9	cart	n02970849	3	handcart	n03484083	2
headdress	n03502509	9	sailing vessel	n04128837	3	warship	n04552696	2
feline	n02120997	8	aircraft	n02686568	3	sled	n04235291	2
amphibian	n01627424	8	bus	n02924116	3	reservoir	n04078574	2
decapod crustacean	n01976146	8	pot	n03990474	3	jar	n03593526	2
place of business	n03953020	8	dish	n03206908	3	basket	n02801938	2
musteline mammal	n02441326	7	pot	n03990474	3	glass	n03438257	2
fungus	n12992868	7	pen	n03906997	3	shaker	n04183329	2
truck	n04490091	7	telephone, phone	n04401088	3	opener	n03848348	2
bottle	n02876657	7	gymnastic apparatus	n03472232	3	power tool	n03997484	2
seat	n04161981	7	neckwear	n03816005	3	pan, cooking pan	n03880531	2
rodent	n02329401	6	swimsuit	n04371563	3	cleaning implement	n03039947	2
mollusk	n01940736	6	body armor	n02862048	3	puzzle	n04028315	2
stringed instrument	n04338517	6	footwear	n03381126	3	camera	n02942699	2
boat	n02858304	6	bridge	n02898711	3	weight	n04571292	2
box	n02883344	6	memorial	n03743902	3	cabinet	n02933112	2
toiletry	n04447443	6	alcohol	n07884567	3	curtain	n03151077	2
bag	n02773037	5	dessert	n07609840	3	sweater	n04370048	2
stick	n04317420	5	cruciferous vegetable	n07713395	3	robe	n04097866	2
bear	n02131653	4	baby bed	n02766320	3	scarf	n04143897	2
woodwind	n04598582	4	procyonid	n02507649	2	gown	n03450516	2
source of illumination	n04263760	4	viverrine	n02134971	2	protective garment	n04015204	2
ship	n04194289	4	cetacean	n02062430	2	oven	n03862676	2
edge tool	n03265032	4	edentate	n02453611	2	sheath	n04187061	2
overgarment	n03863923	4	elephant	n02503517	2	movable barrier	n03795580	2
skirt	n04230808	4	prototherian	n01871543	2	sheet	n04188643	2
roof	n04105068	4	worm	n01922303	2	plaything	n03964744	2
fence	n03327234	4	flower	n11669921	2	mountain	n09359803	2
piece of cloth	n03932670	4	timer	n04438304	2	shore	n09433442	2

Table A. **List of supersets $\mathcal{Y}_{\text{fine}}$.** We list 120 supersets for classes in ImageNet1K. We omit the rest 330 supersets from the list, as they only have single elements (singletons).

	ImageNet (MaxBoxAccV2)				CUB (MaxBoxAccV2)				OpenImages (PxAP)				Total
Methods	VGG	Inception	ResNet	Mean	VGG	Inception	ResNet	Mean	VGG	Inception	ResNet	Mean	Mean
HaS [4]	60.6	**63.7**	63.4	62.6	63.7	53.4	64.6	60.6	58.1	58.1	55.9	57.4	60.2
ACoL [8]	57.4	**63.7**	62.3	61.1	57.4	56.2	66.4	60.0	54.3	57.2	57.3	56.3	59.1
SPG [9]	59.9	63.3	63.3	62.2	56.3	55.9	60.4	57.5	58.3	62.3	56.7	59.1	59.6
ADL [2]	59.9	61.4	63.7	61.6	66.3	58.8	58.3	61.1	58.7	56.9	55.2	56.9	59.9
CutMix [7]	59.5	63.9	63.3	62.2	62.3	57.4	62.8	60.8	58.1	62.6	57.7	59.5	60.8
InCA [3]	61.3	62.8	**65.1**	**63.1**	**66.7**	**60.3**	63.2	63.4	-	-	-	-	-
CAM [10]	60.0	63.4	63.7	62.4	63.7	56.7	63.0	61.1	58.3	63.2	58.5	60.0	61.2
CAM [10] + $\mathcal{Y}$	59.4	62.1	60.4	60.6	63.6	59.1	67.4	63.4	58.3	63.2	58.5	60.0	60.1
CALM_{EM}	62.3	62.2	63.1	62.5	54.9	42.2	60.3	52.5	61.3	64.4	62.5	62.7	59.2
CALM_{EM} + $\mathcal{Y}$	**62.8**	62.3	63.4	62.8	64.8	**60.3**	**71.0**	**65.4**	**61.3**	**64.4**	**62.5**	**62.7**	**63.6**

Table B. **WSOL results on CUB, OpenImages, and ImageNet.** Extension of Table 2 in main paper. CALM_{EM} and CALM_{EM} +$\mathcal{Y}$ are compared against the baseline methods. CALM_{EM} +$\mathcal{Y}$ denote the aggregated attribution map for classes in $\mathcal{Y}$.

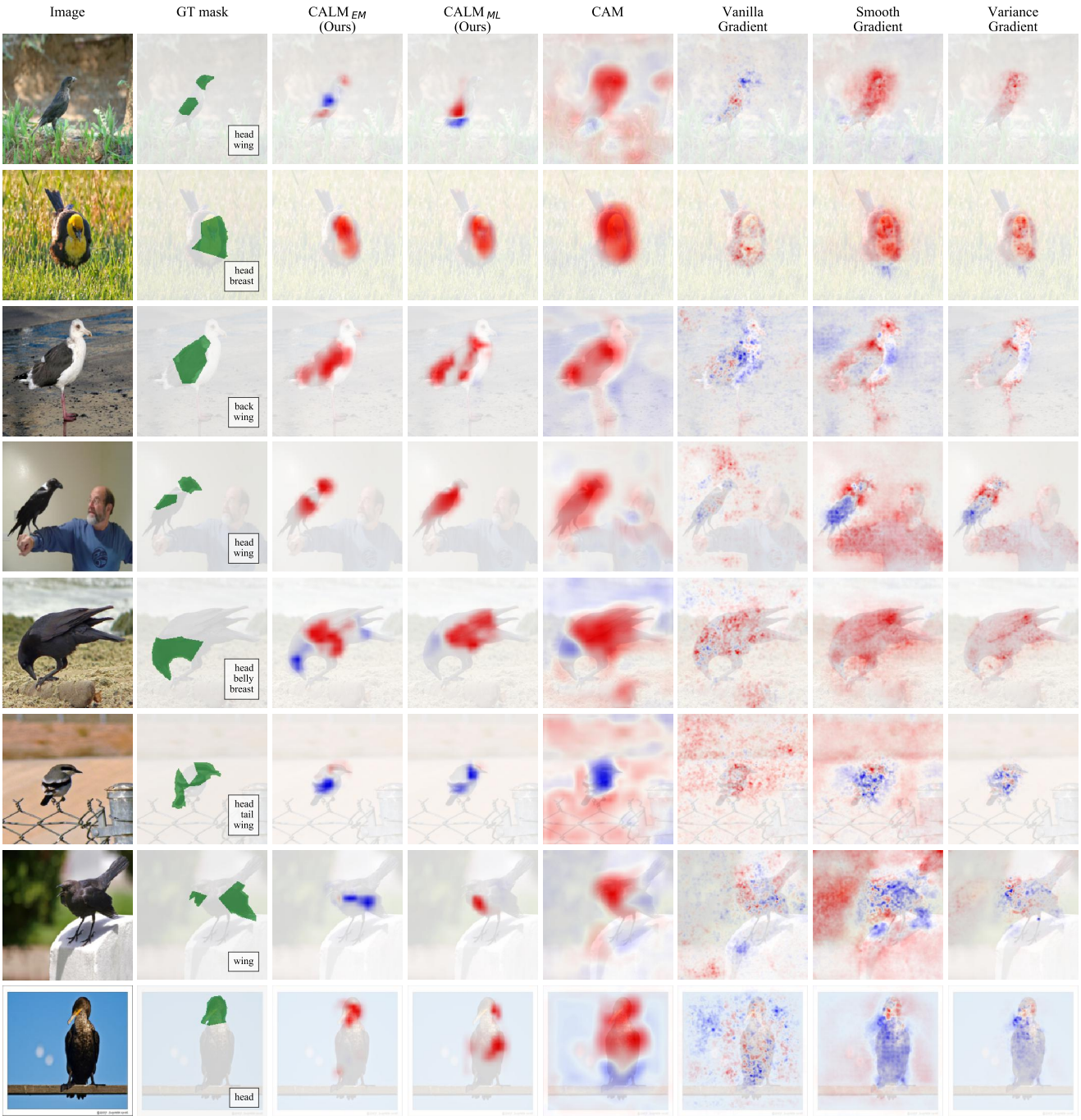

Figure A. **Counterfactual attribution maps on CUB.** Extension of Figure 4 in main paper. We compare the counterfactual attributions from CALM and baseline methods against the GT attribution mask. The GT mask indicates the bird parts where the attributes for the class pair (A,B) differ. The counterfactual attributions denote the difference between the maps for classes A and B: $s^{\mathrm{A}} - s^{\mathrm{B}}$. Red: positive values. Blue: negative values.

Figure B. **Examples of attribution maps on CUB.** We show the object bounding boxes to mark the foreground regions. $+\mathcal{Y}$ denotes the class aggregation.

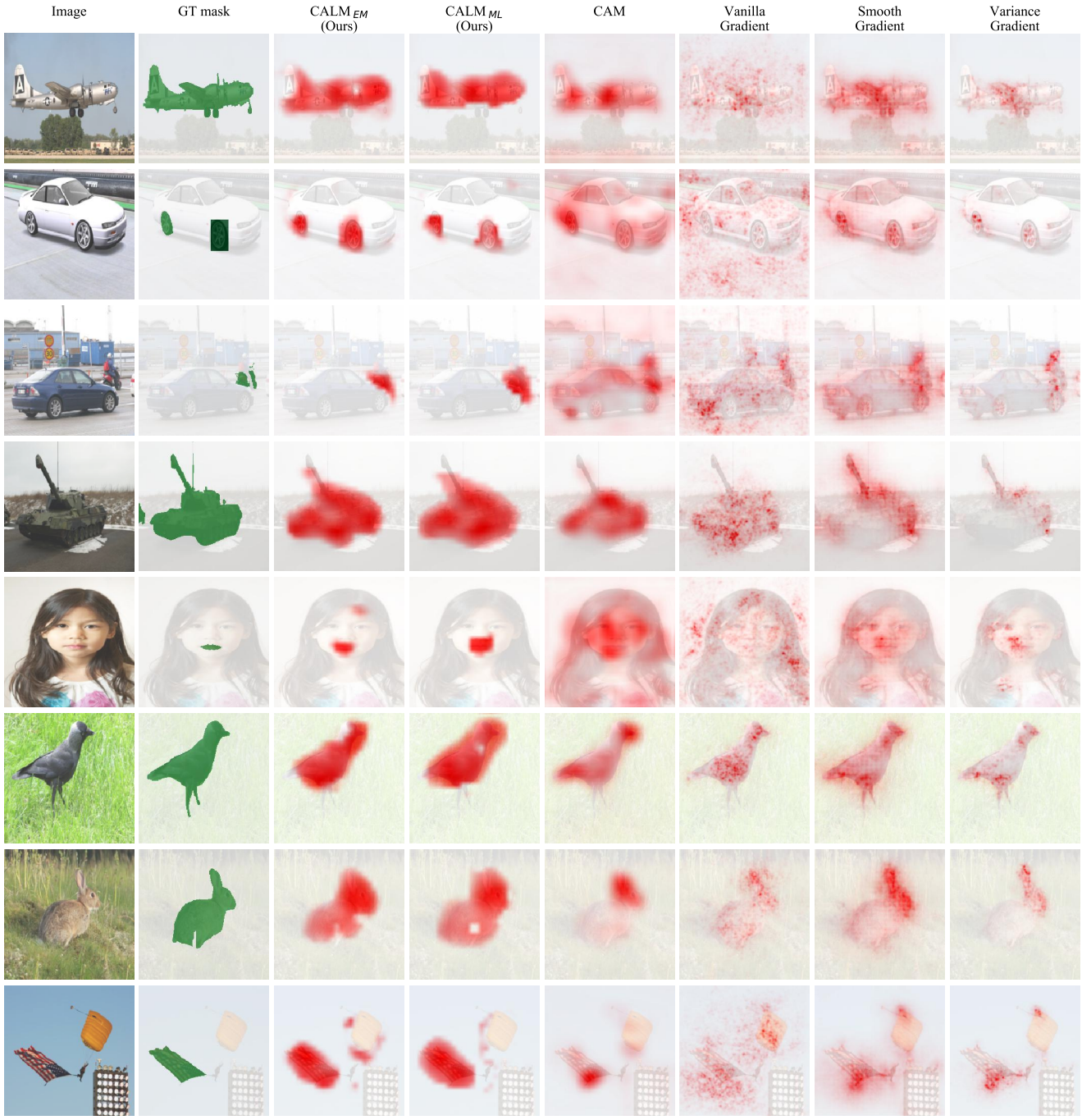

Figure C. **Examples of attribution maps on OpenImages.** We show the object bounding boxes to mark the foreground regions. We do not show the class aggregation (+$\mathcal{Y}$) because it does not change our methods (CALM_{EM} and CALM_{ML}) on OpenImages ($\mathcal{Y}$ are all singletons).

Figure D. **Examples of attribution maps on ImageNet.** We show the object bounding boxes to mark the foreground regions. $+\mathcal{Y}$ denotes the class aggregation.

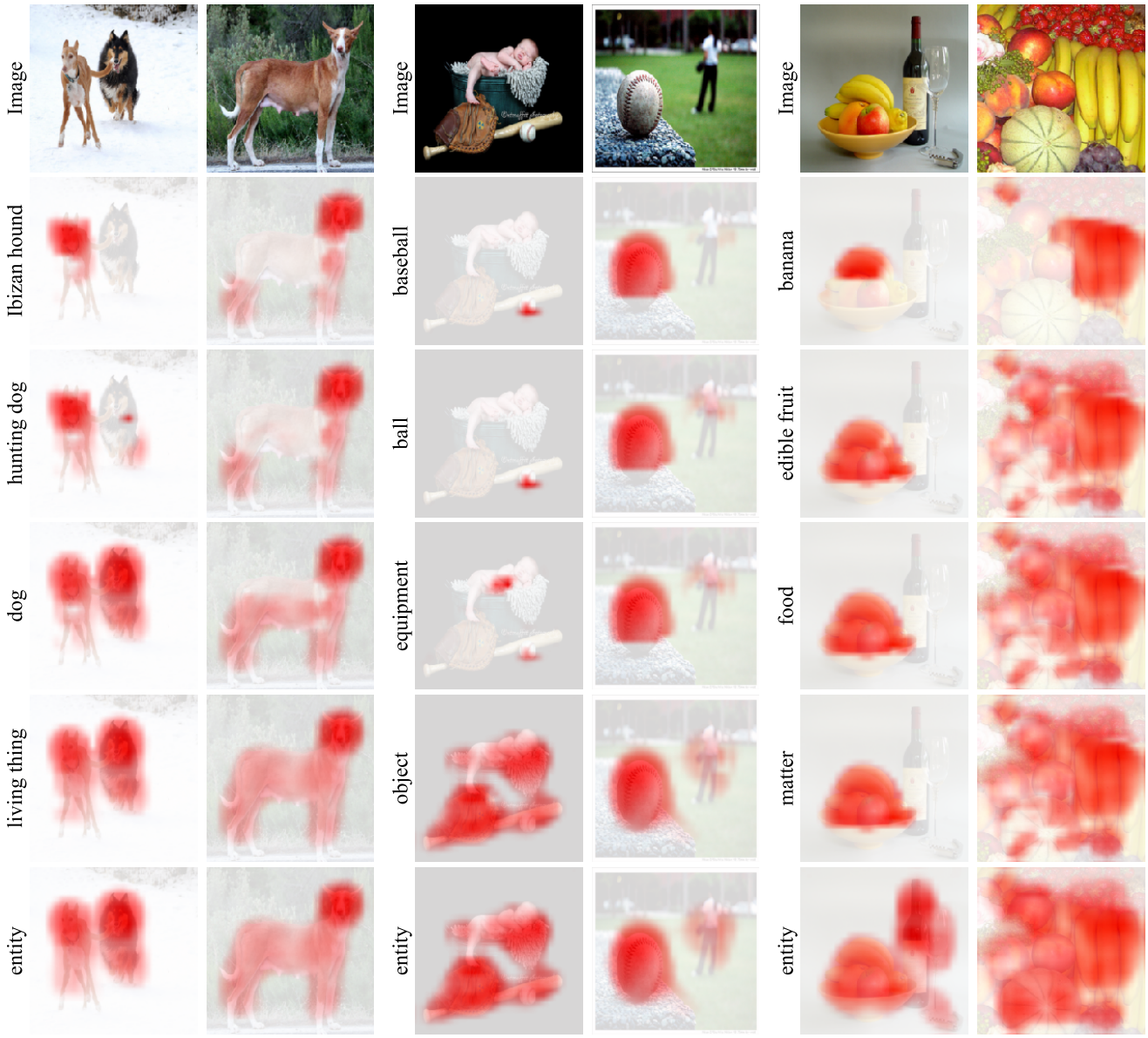

Figure E. **Examples for aggregation at different hierarchies on ImageNet.** Extension of Figure 6 of main paper. We show the aggregated attribution maps at different depths of the hierarchy and the correspondingly expanding superset $\mathcal{Y}$.