Neglected Free Lunch – Learning Image Classifiers Using Annotation Byproducts

Dongyoon Han* NAVER AI Lab
Junsuk Choe* Sogang University
Seonghyeok Chun Dante Company
John Joon Young Chung University of Michigan
Minsuk Chang† NAVER AI Lab
Sangdoo Yun NAVER AI Lab
Jean Y. Song DGIST
Seong Joon Oh University of Tübingen

Abstract

*Supervised learning of image classifiers distills human knowledge into a parametric model f_θ through pairs of images and corresponding labels $\{(X_i, Y_i)\}_{i=1}^N$. We argue that this simple and widely used representation of human knowledge neglects rich auxiliary information from the annotation procedure, such as the time-series of mouse traces and clicks left after image selection. Our insight is that such **annotation byproducts** Z provide approximate human attention that weakly guides the model to focus on the foreground cues, reducing spurious correlations and discouraging shortcut learning. To verify this, we create **ImageNet-AB** and **COCO-AB**. They are ImageNet and COCO training sets enriched with sample-wise annotation byproducts, collected by replicating the respective original annotation tasks. We refer to the new paradigm of training models with annotation byproducts as **learning using annotation byproducts (LUAB)**. We show that a simple multitask loss for regressing Z together with Y already improves the generalisability and robustness of the learned models. Compared to the original supervised learning, LUAB does not require extra annotation costs. ImageNet-AB and COCO-AB are at github.com/naver-ai/NeglectedFreeLunch.*

1. Introduction

Supervised learning of image classifiers requires the transfer of human intelligence to a parametric model f_θ. The transfer consists of two phases. First, human annotators execute human computation tasks [103] to put labels Y on each image X. The resulting labeled dataset $\{(X^i, Y^i)\}_{i=1}^N$ contains the gist of human knowledge about the visual task in a computation-friendly format. In the second phase, the model is trained to predict the labels Y for each input X.

In this work, we question the practice of collecting and

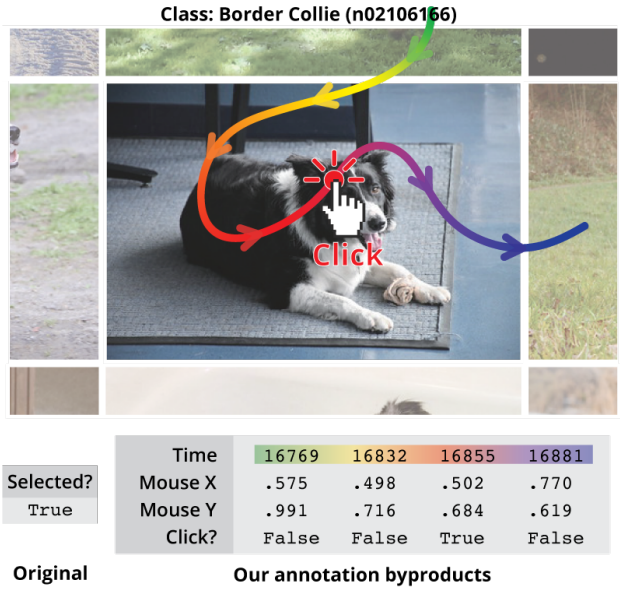


Figure 1: **Annotation byproducts from ImageNet.** Annotators leave traces like click locations as they select images with "Border Collie". We argue that such byproducts contain signals that may improve model generalisation and robustness.

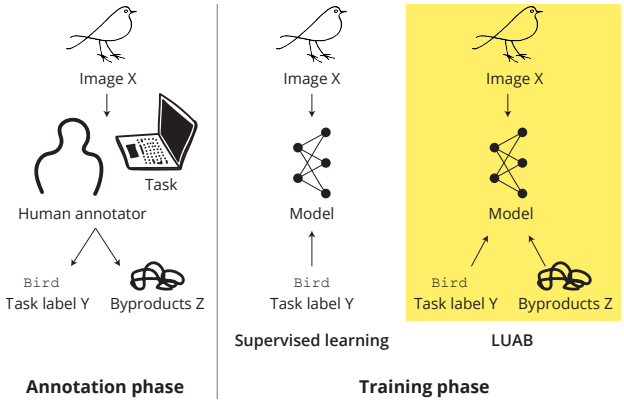


Figure 2: **Learning using Annotation Byproducts (LUAB).** LUAB exploits annotation byproducts Z that are unintentionally generated during the human intelligence tasks for annotation.

*Equal contribution. † currently at Google. Correspondence to Seong Joon Oh: coallaoh@gmail.com.

utilising **only** the labels Y for each image X for training the models. In fact, common practise simply forgoes a large amount of additional signals from human annotators other than mere labels. When humans interact with computers through the graphical user interface, they leave various forms of unintentional traces. Input devices like the computer mouse produce time-series data in which information about what (*e.g.*, mouse action type) and where (*e.g.*, x-y coordinates in the monitor) are logged with timestamps. We refer to such auxiliary signals as **annotation byproducts** Z. See Figure 1 for an ImageNet annotation example [77, 71]. As annotators browse and click on images containing the class of interest, various byproducts are generated, *e.g.*, images over which were hovered during selection, mouse movement speed between images, pixels on which were clicked in an image, images that were deselected due to mistake, and latency between image selections, etc.

We introduce the new learning paradigm, **learning using annotation byproducts (LUAB)**, as a promising alternative to the usual supervised learning (Figure 2). We propose to use the annotation byproducts in the training phase, for further enhancing a model. This is a special case of learning using privileged information (LUPI) [102], where additional information Z other than input X and target Y is available during training but is not given at inference. LUAB is an attractive instance of LUPI, as it does not incur additional annotation costs for privileged information.

We demonstrate the strength of the LUAB framework by contributing datasets **ImageNet-AB** and **COCO-AB**, where the original ImageNet and COCO classification training sets are enriched with the annotation byproducts. We show that annotation byproducts from image-category labelling interfaces contain weak information about the foreground object locations. We show that performing LUAB with such information improves not only generalisability but also robustness by reducing spurious correlations with background features, a critical issue of model reliability these days [87, 54, 28].

Our contributions are (1) acknowledge a neglected information source available without additional costs during image labelling: annotation byproducts (§3); (2) LUAB as a new learning paradigm that makes use of annotation byproducts without extra annotation costs compared to the usual supervised learning (§4); (3) empirical findings that LUAB with byproducts weakly encoding object locations improves model generalisability and reduces spurious correlations with the background (§5); and (4) release of ImageNet-AB and COCO-AB dataset for future research (github.com/naver-ai/NeglectedFreeLunch).

2. Related work

We collect the annotation byproducts of the annotation process and exploit them for training models. We discuss three related fields of machine learning.

2.1. Privileged learning

Privileged learning [100, 101, 102] refers to a machine learning scenario where the model is supervised not only with the directly task-relevant information (*e.g.* image label Y) but also with auxiliary information called **privileged information** (PI) that is not available at inference.

Learning using privileged information (LUPI) was first studied in the context of classical machine learning algorithms such as support vector machines (SVM) [102, 84, 108, 13, 26, 85]. LUPI has since been successfully applied to deep models with multitask learning framework where the PI is plugged in as auxiliary supervision [43, 109, 89]. PI may also be used as a representational bottleneck that regularises the cues for recognition [14, 53, 52]. "Learning with rationale" is an instance of LUPI actively being studied in natural language processing (NLP) domain [11, 46, 33, 110] with recent applications in computer vision problems [91, 36].

Our learning setup, **learning using annotation byproducts (LUAB)**, is an instance of privileged learning with the annotation byproducts as the PI. We hope that LUAB extends the LUPI paradigm by inviting creative methods for utilising the costless annotation byproducts.

2.2. Collecting auxiliary signals from annotators

It has been widely observed in the field of human-computer interaction that online annotators leave traces and logs that contain noisy yet important information [45, 78, 98]. There have been attempts in crowdsourcing image categories to record human gaze during task execution [111, 66, 94, 92, 47, 75, 90, 48]. Since gaze recording devices are costly and intrusive, proxy measurements such as mouse clicks and tracks [6, 69, 7, 62] and partially visible images [18, 51, 50, 56, 57, 24] have also been considered. Other works measure the annotators' response time as a proxy for the sample difficulty [97, 64, 22]. Others have treated the degree of annotator disagreement as the level of difficulty or uncertainty for the sample [83, 67]. Finally, there exist research topics on estimating the annotators' skills and expertise to reflect them in the training phase [9, 86, 80, 58, 95]. In our work, we collect similar signals from annotators, such as mouse signals and interactions with various front-end components. However, our work is the first attempt to collect them at a million scale (*e.g.* ImageNet) that are freely available as byproducts from the original annotation task.

One of the byproducts we collect, namely the click locations during ImageNet annotations, is similar to the "point supervision" considered in some previous work in weakly-supervised computer vision tasks[6, 74, 7]. While the data format (a single coordinate on an image) is similar, those works are *not directly comparable*. Our click locations are *cost-free byproducts* of the original ImageNet annotation procedure that arises *inevitably* from the annotators' selection

of images, while the point supervision requires a dedicated annotation procedure and incurs extra annotation costs.

2.3. Robustness to spurious correlations

Many datasets used for training machine learning models are reported to contain spurious correlations that let the model solve the problem in unintended ways [87, 54, 10, 4, 28, 17, 106]. The presence of such shortcuts is measured through "stress tests" [17]: the model is evaluated against a data distribution where the spurious correlations have been altered or eliminated. We take this approach in §5 to measure improvements in robustness due to LUAB.

Prior approaches to enhance the robustness to spurious correlations have utilised *additional human supervision* to further specify the "correct" correlations models must exploit. For example, [76, 88, 12, 27, 68, 65, 70] regularise the attention maps of image classifiers with respect to various forms of human guidance, such as bounding boxes, segmentation masks, human gaze, and language, to let the classifiers focus on the actual object regions. In this work, we use signals that are *unintentionally* generated by humans during widely-used image annotation procedures to enhance the robustness to spurious correlations. Those signals are available *at no extra cost* during the annotation.

3. Collecting annotation byproducts

To construct a comprehensive package of annotation byproducts, we replicate the annotation procedure for two representative image classification datasets, ImageNet [77], and COCO [55]. Resulting datasets with annotation byproducts, ImageNet-AB and COCO-AB, will be published.

3.1. Browsing versus tagging interfaces

There are two widely-used interfaces for annotating image labels: **browsing** (*e.g.*, ImageNet) and **tagging** (*e.g.*, COCO). A browsing interface presents a single concept along with a set of candidate images arranged in a grid and asks the annotator to select the images correctly depicting the concept. A tagging interface presents a single image at a time and asks the annotator to choose one or more objects and concept labels as necessary (survey of interfaces in [79]).

The two paradigms have different strengths. Browsing is advantageous for efficient batch processing of images, where the annotation precision matters less. Tagging is helpful for careful labelling and supports the annotation of multiple labels per image. Browsing interfaces have been used for the ImageNet [77, 71], Places [116], and CUB [104] datasets. Tagging interfaces have been used for Pascal [23], COCO [55], LVIS [32], and iNaturalist [99]. As representatives of each type, we replicate ImageNet [77, 71] and COCO [55].

3.2. ImageNet

ImageNet [77] is a single-label dataset annotated via browsing. We describe how we replicated the original annotation procedure and present the set of annotation byproducts collected through the browsing annotation.

3.2.1 Replicating ImageNet annotations

We replicate the annotation process for the training split of ImageNet1K (1,281,167 images). The original annotation procedure consists of the following four stages [77, 71]. (1) Construct the list of classes $\mathcal{C}$ to annotate. (2) Crawl candidate images I_c^{cand} for each class $c \in \mathcal{C}$ from the web. (3) Crowdsourced annotators select true images I_c^{select} of class c. (4) Expert annotators clean up the dataset.

We replicate only the crowdsourcing stages (2) and (3) that are directly related to the generation of annotation byproducts. Our replication is based on the description in the original ImageNet [77] and ImageNetV2 [71] papers. For stage (1), we use the 1,000-class subset of the original 21,841 WordNet concepts [63], corresponding to the ILSVRC2012 subset, also known as the ImageNet1K [77].

Preparing candidate images I_c^{cand} for each class $c \in \mathcal{C}$. The candidate images for the original dataset are crawled from Google, MSN, Yahoo, and Flickr [71]. The search keywords are formulated by combining the class names and their "synsets" in WordNet [63]. The resulting set of images I_c^{cand} becomes the candidate image set for class c. The annotators later select a subset $I_c^{\text{select}} \subset I_c^{\text{cand}}$ to finalise the set of images that contain the class c. Our aim is to collect the annotation byproducts for the 1,281,167 original training images of ImageNet1K. We thus let the annotators select the final images from a mixture of the original training images I_c^{imagenet} and the set of new candidate images from Flickr I_c^{flickr} [1]. We set the ratio between the original ImageNet and Flickr-sourced images as 1:3. Our candidate set for each class c is $I_c^{\text{cand}} = I_c^{\text{imagenet}} \cup I_c^{\text{flickr}}$. Then the annotators select the images containing c, $I_c^{\text{select}} \subset I_c^{\text{cand}}$, where the hope is that I_c^{select} contains many original ImageNet samples I_c^{imagenet}. We report 86.7% of I_c^{imagenet} have been selected as a result. A 100% recall is conceptually impossible due to boundary cases and label noises in I_c^{imagenet} [8, 81].

Crowdsourced annotation via browsing interface. Following the original procedure, we let the Amazon Mechanical Turk (MTurk) [2] workers complete the selection process $I_c^{\text{select}} \subset I_c^{\text{cand}}$ for each class c. ImageNet and ImageNetV2 interfaces are shown in Figures 9 and 10 of the ImageNetV2 paper on arXiv [72], respectively. We closely follow the ImageNetV2 interface because the documentation is richer. Our interface is shown in Figure 3. Like ImageNetV2, we show 48 candidate images I_c^{cand} for a single class c for each task. MTurk annotators click on images containing class c and submit the selections I_c^{select}. Importantly, we have designed

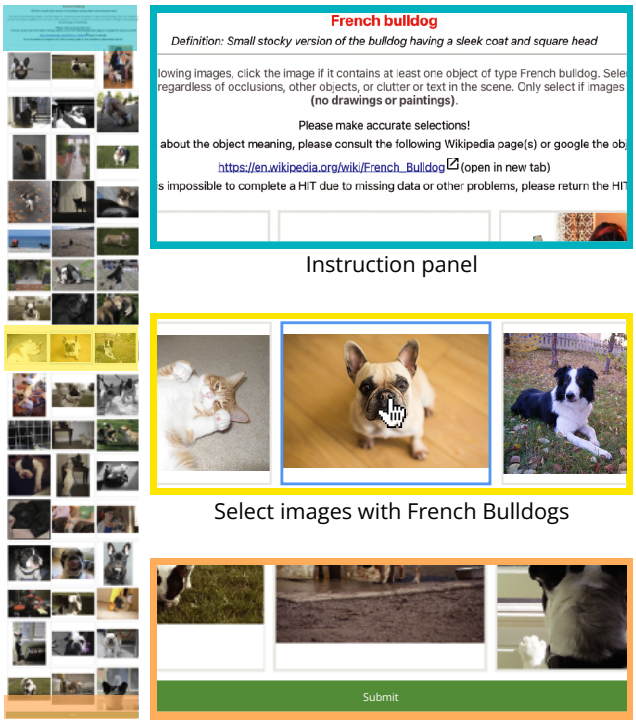


Figure 3: **ImageNet annotation interface.** We replicate the interface in [72]. Annotators read the category description in the instruction panel, select all the images corresponding to "French bulldog", and click on the submit button.

```
"imageID": "n01440764/n01440764_105",
"originalImageHeight": 375,
"originalImageWidth": 500,
"selected": true,                    Original Annotation
"selectedRecord": [
  {"x": 0.540, "y": 0.473, "time": 1641425052}
],
"mouseTracking": [
  {"x": 0.003, "y": 0.629, "time": 1641425051},
  {"x": 0.441, "y": 0.600, "time": 1641425052}
]                                    Annotation Byproducts
```

Figure 4: **Annotation byproducts from ImageNet.** See Appendix Figure A for the full list of byproducts.

the front-end and back-end to record and save the annotation byproducts in the database. The annotation interface and crowdsourcing details are explained in Appendix C.1.

Number of annotators per image. The original ImageNet annotation procedure presents each image to 10 annotators for more precise annotations. This would require 240k USD for the annotation. Given the budget constraint, we have collected 1 annotation per image, spending 24k USD instead. The utility of annotation byproducts demonstrated in §5 is thus *a lower bound on the actual utility*.

3.2.2 ImageNet byproducts

We show the annotation interface for ImageNet in Figure 3. In the ImageNet annotation procedure, annotators click on the images containing the concept of interest. In the process, they leave the time-series of mouse positions (`mouseTracking`) and mouse click events (`selectedRecord`). The original annotation has not recorded them and only saved whether or not each image is finally selected. During our replicated annotation, we saved them in the database. We show the list of annotation byproducts in Figure 4.

Among 1,281,167 ImageNet1K training images, annotators re-selected 1,110,786 (86.7%) and interacted with 1,272,225 (99.3%) images, leaving annotation byproducts.

3.3. COCO

COCO [55] is a multi-label dataset annotated with a tagging interface. We describe the creation of COCO-AB. We present and analyse the annotation byproducts for COCO.

3.3.1 Replicating COCO annotations

We replicate annotations for the 82,783 training images of COCO 2014 to collect the annotation byproducts. The original annotation procedure for COCO [55] consists of four stages. (1) Construct a list of classes to annotate. (2) Crawl and select candidate images from Flickr with more emphasis on images with multiple objects in context. (3) For each image, let crowdsourced annotators put all valid category labels. (4) Expert annotators do a final check-up.

We only replicate stage (3), which produces direct annotation byproducts, by letting annotators work on the 82,783 training images. Figure 5 shows the COCO annotation interface. We replicate the front-end of the original [55] (Figure 12a). For every image presented, the annotator must identify as many classes present as possible and place the corresponding class icons on the objects. We have replicated the superclass-browsing interface in [55] that lets annotators efficiently search through 80 COCO classes via 11 superclasses. The icon can be placed only once on an image per class. That is, even when there are multiple instances of a class, annotators should choose one of them to place the icon on. This is the same in the original COCO interface. Crowdsourcing details are in Appendix C.2.

3.3.2 COCO Byproducts

COCO interface (Figure 5) has two main components: (1) the image on which the class icons are placed and (2) the class browsing tool showing the class icons. The annotation byproducts come from these two sources. See Figure 6 for the full list of annotation byproducts.

The `actionHistories` field describes the actions performed with the mouse cursor on the image. It lists the sequence of actions with possible types `add`, `move`, `remove` and the corresponding location, time, and the category label of the icon. The `mouseTracking` field records the movement of the mouse cursor over the image.

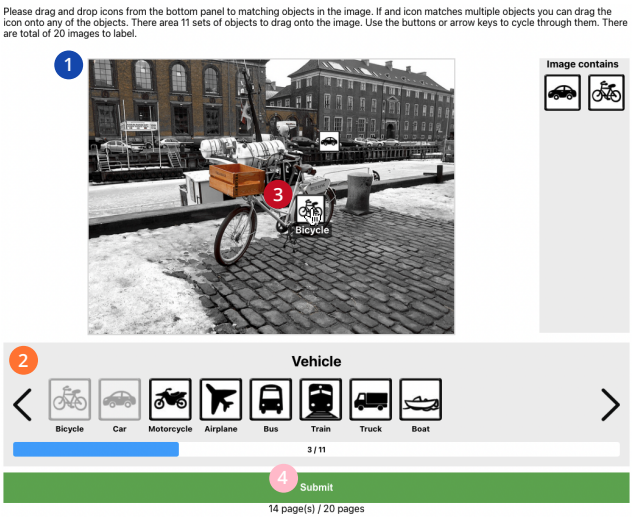


Figure 5: **COCO annotation interface.** 1 Annotator works on a single image at a time. 2 Find the classes present in the image by navigating superclasses. 3 Drag and drop class icons on the objects in the image. 4 When finished, click on the submit button.

```
"image_id": 459214,
"originalImageHeight": 428,
"originalImageWidth": 640,
"categories": ["car", "bicycle"], Original Annotation
"actionHistories": [
  {"actionType": "add",
   "iconType": "car",
   "pointTo": {"x": 0.583, "y": 0.588},
   "timeAt": 16686},
  {"actionType": "add",
   "iconType": "bicycle",
   "pointTo": {"x": 0.592, "y": 0.639},
   "timeAt": 16723}
],
"mouseTracking": [
  {"x": 0.679, "y": 0.862, "timeAt": 15725},
  {"x": 0.717, "y": 0.825, "timeAt": 15731}
]                                  Annotation Byproducts
```

Figure 6: **Annotation byproducts from COCO.** See Appendix Figure B for the full list of byproducts.

Annotators have reannotated 82,765 (99.98%) of the 82,783 training images. We found that only 61.9% of the class occurrences are retrieved on average. This confirms the findings in Lin *et al*. [55] that the recall rate is low for multi-label annotation tasks and multiple annotators are necessary for every image. While desirable, collecting 10 annotations per image requires 100k USD, beyond our budget. We have instead assigned one annotator per image, spending 10k USD. Our setup presents a lower bound on the actual utility of the original annotation byproducts.

Finally, we emphasise those localisation byproducts are indeed general annotation byproducts for class labelling with a tagging interface. For example, Objects365 classes are obtained by labelling the 365 classes *along with instance bounding boxes* (§3.2.1 in [82]). Class labels in LVIS are collected *along with corresponding positions*, as in COCO (§3.1 in [32]). Location marking is often inseparable from multi-label annotations. Without any indication of *where*, subsequent quality control stages are highly inefficient. Suppose an annotator labels "chopsticks" in a cluttered kitchen photo. It will be challenging to quickly confirm if the label is correct without knowing *where*.

4. Learning using annotation byproducts

We introduce the paradigm of **learning using annotation byproducts (LUAB)**. Compared to conventional supervised learning, we train models with additional annotation byproducts that have previously not been utilised in model training.

4.1. LUAB with weak localisation signals

Annotation byproducts contain rich information surrounding the input image and the cognitive process of the annotator executing the task. In this work, we focus on the byproducts related to **object locations**, such as the click locations on images. We expect them to provide the model with a weak signal on the actual foreground pixels of the objects. Albeit weak, we expect them to be helpful information for resolving spurious correlations with background features, a common phenomenon in vision datasets [107, 87].

Annotation byproducts encoding object locations. We hypothesise that the record of human interaction with the image annotation interfaces provides weak signals for the object locations. For ImageNet (§3.2), we consider the final click coordinates for every selected image (`selectedRecord`). For COCO (§3.3), we consider the coordinates of the final `add` action of a class icon on the image (`actionHistories`). We treat them as proxy, cost-free data for object locations for each image. We note that such points on objects provide rich information about the foreground locations [6, 7].

Precision of object localisation in annotation byproducts. We verify the localisation accuracy of the annotation byproducts mentioned above. For ImageNet, we consider the subset of training data with both (1) our annotation byproducts (87%) and (2) ground-truth boxes provided by the original dataset (42%). We use the boxes to measure click accuracy. This gives 82.9% accuracy. Qualitative examples are in Figure 7. For COCO, we use the ground-truth pixel-wise masks for measuring the precision of icon placements (#correct placement/#all placements). This gives 92.3% precision. Therefore, we confirm that the respective annotation byproducts are fairly precise proxies for the actual foreground pixels. See Appendix E for more analysis.

Other annotation byproducts from class labelling. We conjecture that one may obtain an estimate for the extent of objects by taking the convex hull of a few mouse trajectory points before and after the click or icon placement. In addition to localisation, annotation byproducts may provide proxy signals on sample-wise difficulty through the completion time [97]. There also exists rich cross-sample association information: where two samples are annotated

by the same annotator or on the same front-end page. Such information may help reduce annotator biases [29]. They are beyond the scope of our paper, but we discuss the possibilities in Appendix §D.1.

Annotation byproducts from tasks beyond class labelling. Polygonal instance segmentation [55] results in byproducts like the order of clicks and the history of corrections. In the language domain, one may not only record human text answers but the history of corrections in the answer, where we hypothesise that more corrections signify more ambiguity.

4.2. Multi-task learning baseline for LUAB

The usual ingredients for the supervised learning of image classifiers are image-label pairs (X, Y). Our LUAB framework introduces a third ingredient, weak object location Z, for every image X. For single-class datasets like ImageNet, the coordinates are given as $Z \in [0,1] \times [0,1]$, a relative position in each image. For multi-class datasets like COCO, this is given as $Z_c \in [0,1] \times [0,1]$ for every class c present in the image.

We propose a simple baseline based on a **multi-task objective** for the classification of Y and the regression of Z. We expect that learning the localisation would condition the network to select features more from foreground object regions [113, 61, 25].

We write the original network architecture as $g(f(X))$, where f is a feature extractor, and g is a classifier that maps intermediate features to $\mathbb{R}^C$. The regression objective is applied to $h(f(X))$ where h maps the intermediate features to normalised x-y coordinates in $[0,1] \times [0,1]$. For a single-class classification task (*e.g.* ImageNet), the objective is

$$\min_{f,g,h} \mathcal{L}\left(g(f(X)), Y\right) + \lambda ||h(f(X)) - Z||_{s1}, \qquad (1)$$

where $\mathcal{L}$ is the cross-entropy loss and $||\cdot||_{s1}$ is the smooth-ℓ^1 loss [30]. $\lambda > 0$ regulates the weight of the regression term. The objective is identical for the multi-class classification (*e.g.* COCO), except that $\mathcal{L}$ is a binary cross-entropy loss and the regression target is the mean of smooth-ℓ^1 losses for every class present in the image. We use the task labels Y from the original datasets for both ImageNet and COCO experiments. The regression term is applied only for samples for which Z is available.

Discussion. We show the minimal utility of the annotation byproducts by considering a simple baseline. We note that one may explore more advanced training schemes like regularising the model's attribution map with Z [76, 88, 12] or forcing the model to pool features with attention Z [14]. We explore the latter method in Appendix §F.

Figure 7: **ImageNet final clicks**. We visualise random training images; **points** are the final click positions in `selectedRecord`.

5. Experimental results

We show the empirical efficacy of **learning using annotation byproducts (`LUAB`)** that weakly encode object locations. We verify whether the annotation byproducts improve the original image classification performance and robustness by guiding models to focus more on foreground features.

5.1. Results on ImageNet

Implementation details. We use the ImageNet-AB training set with annotation byproducts to train image classifiers. Considered backbones are ResNets [35] (ResNet18, ResNet50, ResNet101, and ResNet152), and Vision Transformers (ViT-Ti [96], ViT-S [96], and ViT-B [21]). To accommodate the multi-task objective, we have attached a separate head for the regression target at the penultimate layer of each backbone. This head is not used during the inference. We use the standard 100-epochs setup [35] for ResNets; the DeiT training setup[1] [96] is used for ViTs. This is to verify whether the annotation byproducts work together with the popular supervised training regimes. We select the last-epoch models. We further include results following the primitive setup [21] in Appendix Table B.

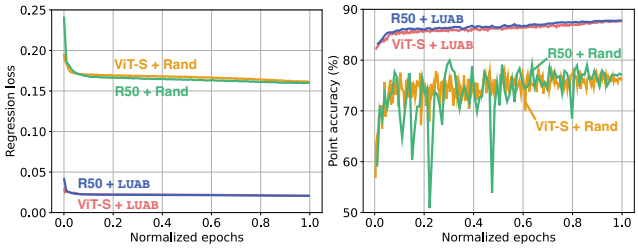
Evaluation. Along with the ImageNet1k validation set (IN-1k), we use many variants: ImageNet-V2/Real/A/C/O/R/Sketch/ObjNet [71, 8, 39, 38, 105, 37, 5]. In particular, we focus on the benchmarks designed to measure spurious correlations with the background cues: SI-Score [20] and BG Challenge [107]. Both datasets de-correlate the foreground and background features by constructing novel images with foreground and background masks cut and pasted from different images.

Random point baseline. We introduce a baseline trained with the same objective (Equation 1) but with a uniform-random point Z for each image. This baseline helps us rule out possible regularisation effects due to the multi-task learning itself and focus purely on the information gain from the

[1] We train models with the official DeiT codebase [96] with default settings for RandAug [16], Stochastic Depth [44], Random Erasing [41, 19], Mixup [114], Cutmix [112], and optimization setups – AdamW [60] and cosine learning rate scheduling [59], and gradual warmup [31].

Model	Params	IN-1k↑	IN-V2↑	IN-Real↑	IN-A↑	IN-C↑	IN-O↑	Sketch↑	IN-R↑	Cocc↑	ObjNet↑	SI-size↑	SI-loc↑	SI-rot↑	BGC-gap↓	BGC-acc↑
R18	11.7M	72.1	59.9	**74.6**	**2.0**	37.4	52.7	**22.0**	34.0	41.9	21.7	46.4	22.9	32.1	9.0	**22.1**
+LUAB	11.7M	**72.2**	**59.9**	74.5	1.9	**37.6**	**53.0**	21.6	**34.3**	**44.7**	**21.9**	**47.8**	**23.1**	**32.7**	**8.6**	20.4
R50	25.6M	77.4	65.2	78.2	**5.5**	43.8	56.7	25.4	37.8	53.7	27.8	53.9	31.9	40.1	6.3	26.7
+LUAB	25.6M	**77.5**	**65.2**	**78.5**	5.1	**44.7**	**57.0**	**25.7**	**38.2**	**55.1**	**28.5**	**55.6**	**33.5**	**40.9**	**5.6**	**27.4**
R101	44.5M	78.2	66.0	78.8	7.6	47.0	**60.7**	26.5	38.2	55.8	29.4	53.4	33.1	38.9	5.6	**30.2**
+LUAB	44.5M	**78.6**	**66.4**	**79.0**	**7.8**	**47.9**	60.5	**27.0**	**39.0**	**58.5**	**30.0**	**54.4**	**33.3**	**39.8**	**5.5**	28.2
R152	60.2M	79.0	67.2	79.2	9.5	49.5	62.0	27.6	39.6	58.8	30.5	53.9	33.3	38.6	6.6	27.2
+LUAB	60.2M	**79.2**	**67.2**	**79.4**	**9.5**	**49.9**	**62.1**	**27.6**	**39.7**	**59.0**	**31.3**	**55.5**	**34.2**	**40.6**	**5.8**	**31.6**
ViT-Ti	5.7M	72.8	60.7	75.6	7.9	**48.5**	52.3	20.5	32.8	63.8	23.1	46.3	23.8	33.9	8.2	13.9
+LUAB	5.7M	**72.9**	**60.8**	**75.8**	**8.4**	48.4	**52.9**	**21.1**	**33.8**	**64.2**	**23.7**	**47.4**	**25.4**	**34.7**	**7.8**	**14.4**
ViT-S	22.1M	80.3	69.1	80.6	20.0	60.3	53.4	29.4	42.3	73.8	31.2	54.5	32.0	39.5	6.4	17.4
+LUAB	22.1M	**80.6**	**69.7**	**81.0**	**22.8**	**61.2**	**55.1**	**30.0**	**43.0**	**74.1**	**32.3**	**55.1**	**33.7**	**39.6**	**5.9**	**18.7**
ViT-B	86.6M	81.6	70.3	81.1	26.1	64.1	58.0	33.0	45.7	76.0	31.7	56.6	35.1	41.3	6.4	18.1
+LUAB	86.6M	**82.5**	**71.9**	**81.8**	**31.1**	**66.0**	**58.5**	**35.5**	**48.4**	**77.5**	**35.0**	**57.1**	**36.8**	**41.6**	**5.6**	**23.9**

Table 1: **Performance of LUAB on ImageNet1K.** We report in-distribution generalisation metrics (IN-1k/V2/Real) and out-of-distribution metrics (IN-A/C/O/R/Sketch/Cocc/ObjNet). We also report metrics for detecting spurious correlations with background (SI-Score [20] and BG-Challenge [107]). LUAB training with annotation byproducts using a simple point regression target improves the overall performances. LUAB barely introduces any extra annotation or computational cost.



(a) Training reg. loss (b) Validation loc. accuracy

Figure 8: **Training curves for ImageNet**. "Rand" refers to the regression with respect to a randomly generated location Z.

Model	Annot.	IN-1k↑	ObjNet↑	SI-size↑	SI-loc↑	SI-rot↑	BGC-gap↓	BGC-acc↑
R50	-	77.4	27.8	53.9	31.9	40.1	6.3	26.7
R50	Rand	77.3	28.1	54.5	31.5	39.7	5.9	**27.6**
R50	LUAB	**77.5**	**28.5**	**55.6**	**33.5**	**40.9**	**5.6**	27.4
ViT-Ti	-	71.8	20.1	40.6	16.5	26.2	12.1	13.6
ViT-Ti	Rand	72.2	22.0	42.5	18.1	27.5	11.0	15.3
ViT-Ti	LUAB	**73.0**	**22.1**	**43.4**	**20.0**	**28.7**	**10.9**	**16.1**
ViT-S	-	74.1	20.5	42.9	18.7	27.8	10.5	16.7
ViT-S	Rand	74.8	22.7	44.5	20.6	28.8	10.5	19.5
ViT-S	LUAB	**75.3**	**23.6**	**47.8**	**22.6**	**32.2**	**8.7**	**19.7**

Table 2: **Comparison with random point regression on ImageNet.** We compare the accuracies of supervised learning without additional supervision ("-"), with random points as guidance ("Rand"), and with our annotation byproducts (LUAB).

weak object locations given by the annotation byproducts.

LUAB trains well. Figure 8 shows the training curves. The regression loss for Z decreases and the validation localisation accuracy increases for LUAB over the epochs. The baseline random-point supervision experiments yield higher loss values and lower validation localisation accuracy. The baseline performance is still fairly high, as ImageNet is object-centric. We confirm that the annotation byproducts contain localisation information that lets the model predict object locations.

LUAB improves classification performance. See Table 1 for the IN-1k validation accuracies before and after LUAB. We observe that LUAB introduces gains across the board (*e.g.* 81.6% to 82.5% for ViT-B). Similar gains are seen for IN-V2/Real. The LUAB help the models generalise better.

LUAB improves out-of-distribution (OOD) generalisation. Table 1 shows that LUAB improves the OOD generalisation (columns for IN-A/C/O/R/Sketch). 30 of the 35 combinations (5 metrics $\times$ 7 models) have seen improvements due to LUAB. We hypothesise that the focus on foreground features improves generalisation to novel distributions.

LUAB reduces spurious correlations with the background. Table 1 also shows the results on metrics detecting spurious dependence on background features. For SI-Scores [20], we observe a clear advantage of LUAB, beating the baseline performance in *all* considered cases. For BG Challenge [107], LUAB surpasses the original models for the majority of cases (12 out of 14). The improvement due to LUAB on the benchmarks with de-correlated foreground and background features demonstrates the efficacy of the foreground guidance from the annotation byproducts.

Improvement is not due to the multi-task objective itself. Table 2 shows greater improvements due to LUAB compared to the random point baseline, which merely introduces a multi-task learning objective without additional location information. As such, we attribute the improvements to the

Annot.	Loc↑
R50	46.8
+LUAB	**48.4**

Table 3: **WSOL on ImageNet** [15].

Annot.	IN-1k↑	Bbox AP↑	Mask AP↑
R50	77.4	37.0	34.6
+LUAB	**77.5**	**37.4**	**34.8**

Table 4: **Fine-tuning ImageNet models on downstream tasks.** Object detection and instance segmentation.

weak foreground information in the annotation byproducts.

LUAB lets models focus on foreground features. Class activation mapping (CAM) [115] identifies the region-wise features that an image classifier uses to make the prediction. By using a weakly-supervised object localisation (WSOL) evaluation against the ground-truth object locations [15], one may confirm whether the utilised image features correspond to the object foreground. We show the results in Table 3. The 1.6%p improvement in WSOL accuracy against the original shows that LUAB lets the model focus on the foreground.

LUAB improves downstream localisation tasks. We report the box and mask APs on COCO val2017 after fine-tuning the baseline ResNet50 and LUAB-trained models for Faster-RCNN [73] and Mask-RCNN [34], respectively, in Table 4. LUAB improves the downstream performances.

5.2. Results on COCO

Implementation details. We use the COCO-AB training set with annotation byproducts. Considered backbones are ResNet18/50/152 [35], and ViT-Ti/S/B [96, 21]. We attach one regression head per class on the penultimate layer. We follow the training recipe of the original papers. As in ImageNet, we consider the random point baseline, where the localisation supervision Z_c is given as a uniform-random point.

LUAB trains well. Figure 9 shows the training curves for COCO with LUAB. Compared to the random-point baseline, LUAB decreases the regression loss and increases the validation localisation accuracy more quickly. We confirm: LUAB confers the model information about where the objects are.

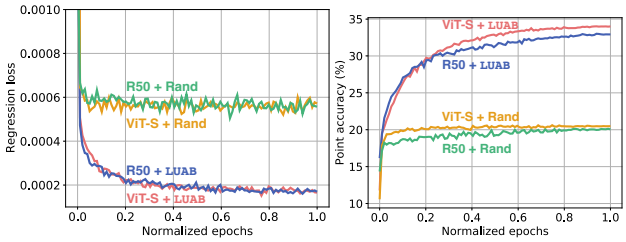
LUAB improves classification performance. Table 6 and 7 show that LUAB improves the mean average precision (mAP), for example from 73.0% to 74.2% for ResNet50.

LUAB reduces spurious correlations with other classes. We consider metrics for detecting a spurious dependence on frequently co-occurring objects (*e.g.* monitor and keyboard). V^{avg} and V^{min} [87] compute the difference between the classification scores when class c of interest is removed and when another class than c are removed. V^{avg} erases a random class, while V^{min} erases the worst-case class for each image. Table 6 and 7 show a consistent decrease in V^{avg} and V^{min} scores after LUAB. This confirms the successful reduction in spurious background correlations via LUAB.

LUAB lets models focus on foreground features. As in ImageNet, we measure the CAM performances of the COCO-trained ResNet50 with and without LUAB in Table 5. We compute CAM for every class and report the class-averaged mPxAP [15]. We verify that the models attend more to the foreground features after training with LUAB.

Annot.	mPxAP↑
R50	20.8
+LUAB	**21.5**

Table 5: **WSOL on COCO** [15].



(a) Training reg. loss (b) Validation loc. accuracy

Figure 9: **Training curves for COCO**. "Rand" refers to the regression with respect to randomly generated locations Z_c.

Model	R18	Rand	LUAB	R50	Rand	LUAB	R152	Rand	LUAB
mAP↑	67.9	67.8	**68.0**	73.0	73.6	**74.2**	73.3	74.6	**75.4**
V^{min}↓	51.8	52.1	**51.6**	47.6	47.3	**47.0**	47.4	47.8	**47.1**
V^{avg}↓	28.7	28.7	**28.4**	25.0	24.9	**24.5**	24.8	25.5	**24.7**

Table 6: **COCO Performance with ResNet.** We compare supervised learning, multi-task learning with random points, and LUAB.

Model	ViT-Ti	Rand	LUAB	ViT-S	Rand	LUAB	ViT-B	Rand	LUAB
mAP↑	72.6	72.2	**72.7**	76.2	76.9	**77.3**	76.4	74.5	**77.5**
V^{min}↓	49.1	48.9	**48.4**	47.1	46.9	**45.8**	46.6	47.1	**45.6**
V^{avg}↓	27.0	26.9	**26.8**	25.7	25.6	**24.6**	25.0	25.1	**24.5**

Table 7: **COCO Performance with ViT.** We compare supervised learning, multi-task learning with random points, and LUAB.

6. Conclusion

We propose to log and exploit annotation byproducts that result from human interaction with input devices and various front-end components. We have created **ImageNet-AB** and **COCO-AB** by replicating the respective annotation procedures and logging **cost-free** annotation byproducts. We have introduced a new learning paradigm: **learning using annotation byproducts (LUAB)**. As an example, we have used the final click and icon placement locations as proxies for the object locations. They let models generalise better and depend less on spurious background features.

Limitations. We have performed only one annotation pass through ImageNet and COCO, rather than the 10× repetitions done in the original procedure. We may have seen even stronger results with LUAB if annotation byproducts were collected during the original procedure. There are also exciting possibilities for exploiting other types of byproducts; one may also estimate image difficulty and annotator biases from the raw annotation byproducts. Finally, we have restricted our scope to image classifiers. We believe that the LUAB paradigm will benefit other tasks and domains, such as text, audio, video, and tabular data.

Take-home messages for dataset building. When building

a dataset, one should consider logging and releasing the annotation byproducts, along with the main annotations. They may improve models' generalisation and robustness for free.

Ethical concerns. Our data collection for ImageNet-AB and COCO-AB has obtained an IRB approval from an author's institute. We note that there exist potential risks that annotation byproducts may contain annotators' privacy. Data collectors may even attempt to leverage more private information as byproducts. We urge data collectors not to collect or exploit private information from annotators. Whenever appropriate, one must ask for the annotators' consent.

Acknowledgements. We are grateful to NAVER and DGIST for funding the MTurk annotations. We credit Kay Choi for designing the figures. We thank Elif Akata, Elisa Nguyen, and Alexander Rubinstein for reviewing the manuscript. Experiments are based on the NSML [49] platform.

References

[1] Flickr. https://flickr.com, 2004. 3

[2] Amazon mechanical turk. https://www.mturk.com/, 2005. 3

[3] Us federal minimum wage. https://www.dol.gov/general/topic/wages/minimumwage, 2022. 14

[4] Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning de-biased representations with biased representations. In *International Conference on Machine Learning*, 2020. 3

[5] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits object recognition models. *Advances in Neural Information Processing Systems*, 2019. 6

[6] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *European Conference on Computer Vision*, pages 549–565. Springer, 2016. 2, 5, 17

[7] Rodrigo Benenson and Vittorio Ferrari. From couloring-in to pointillism: revisiting semantic segmentation supervision. In *ArXiv*, 2022. 2, 5

[8] Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with imagenet? *arXiv preprint arXiv:2006.07159*, 2020. 3, 6

[9] Steve Branson, Grant Van Horn, and Pietro Perona. Lean crowdsourcing: Combining humans and machines in an online system. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7474–7483, 2017. 2

[10] Remi Cadene, Corentin Dancette, Matthieu Cord, Devi Parikh, et al. Rubi: Reducing unimodal biases for visual question answering. In *Advances in Neural Information Processing Systems*, pages 839–850, 2019. 3

[11] Samuel Carton, Surya Kanoria, and Chenhao Tan. What to learn, and how: Toward effective learning from rationales. *arXiv preprint arXiv:2112.00071*, 2021. 2

[12] Hila Chefer, Idan Schwartz, and Lior Wolf. Optimizing relevance maps of vision transformers improves robustness. *arXiv preprint arXiv:2206.01161*, 2022. 3, 6

[13] Jixu Chen, Xiaoming Liu, and Siwei Lyu. Boosting with side information. In *Asian Conference on Computer Vision*, pages 563–577. Springer, 2012. 2

[14] Yunpeng Chen, Xiaojie Jin, Jiashi Feng, and Shuicheng Yan. Training group orthogonal neural networks with privileged information. *arXiv preprint arXiv:1701.06772*, 2017. 2, 6

[15] Junsuk Choe, Seong Joon Oh, Seungho Lee, Sanghyuk Chun, Zeynep Akata, and Hyunjung Shim. Evaluating weakly supervised object localization methods right. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3133–3142, 2020. 8

[16] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical data augmentation with no separate search. *arXiv preprint arXiv:1909.13719*, 2019. 6, 19, 21

[17] Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. Underspecification presents challenges for credibility in modern machine learning. *arXiv preprint arXiv:2011.03395*, 2020. 3

[18] Jia Deng, Jonathan Krause, and Li Fei-Fei. Fine-grained crowdsourcing for fine-grained recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2013. 2

[19] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 6, 19, 21

[20] Josip Djolonga, Jessica Yung, Michael Tschannen, Rob Romijnders, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Matthias Minderer, Alexander D'Amour, Dan Moldovan, et al. On robustness and transferability of convolutional neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16458–16468, 2021. 6, 7

[21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, and Sylvain Gelly. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 6, 8, 19, 20, 21

[22] Justin Dulay and Walter J Scheirer. Using human perception to regularize transfer learning. *arXiv preprint arXiv:2211.07885*, 2022. 2

[23] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 3

[24] Thomas Fel, Ivan Felipe, Drew Linsley, and Thomas Serre. Harmonizing the object recognition strategies of deep neural networks with humans. In *Advances in Neural Information Processing Systems*, 2022. 2

[25] Chris Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. Efficiently identifying task groupings for multi-task learning. *Advances in Neural Information Processing Systems*, 34:27503–27516, 2021. 6

[26] Shereen Fouad, Peter Tino, Somak Raychaudhury, and Petra Schneider. Incorporating privileged information through

metric learning. *IEEE Transactions on Neural Networks and Learning Systems*, 24(7):1086–1098, 2013. 2

[27] Yuyang Gao, Tong Steven Sun, Liang Zhao, and Sungsoo Ray Hong. Aligning eyes between humans and deep neural network through interactive attention alignment. *Proceedings of the ACM on Human-Computer Interaction*, 6(CSCW2):1–28, 2022. 3

[28] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. 2, 3

[29] Mor Geva, Yoav Goldberg, and Jonathan Berant. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. *arXiv preprint arXiv:1908.07898*, 2019. 6

[30] Ross Girshick. Fast R-CNN. In *International Conference on Computer Vision*, pages 1440–1448, 2015. 6

[31] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 6

[32] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 3, 5

[33] Braden Hancock, Martin Bringmann, Paroma Varma, Percy Liang, Stephanie Wang, and Christopher Ré. Training classifiers with natural language explanations. In *Association for Computational Linguistics Meeting*, 2018. 2

[34] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *IEEE international conference on computer vision*, pages 2961–2969, 2017. 8

[35] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016. 6, 8, 19

[36] Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. Generating visual explanations. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *European Conference on Computer Vision*, pages 3–19, 2016. 2

[37] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021. 6

[38] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 6

[39] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271, 2021. 6

[40] Byeongho Heo, Sanghyuk Chun, Seong Joon Oh, Dongyoon Han, Sangdoo Yun, Gyuwan Kim, Youngjung Uh, and Jung-Woo Ha. Adamp: Slowing down the slowdown for momentum optimizers on scale-invariant weights. In *International Conference on Learning Representations*, 2020. 19

[41] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. 6, 19, 21

[42] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8129–8138, 2020. 19

[43] Judy Hoffman, Saurabh Gupta, and Trevor Darrell. Learning with side information through modality hallucination. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 826–834, 2016. 2

[44] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Weinberger. Deep networks with stochastic depth. In *European Conference on Computer Vision*, 2016. 6, 19, 21

[45] Dietmar Janetzko. Nonreactive data collection. *The SAGE Handbook of Online Research Methods*, pages 161–173, 2008. 2

[46] Sahil Jayaram and Emily Allaway. Human rationales as attribution priors for explainable stance detection. In *2021 Conference on Empirical Methods in Natural Language Processing*, pages 5540–5554, 2021. 2

[47] Nour Karessli, Zeynep Akata, Bernt Schiele, and Andreas Bulling. Gaze embeddings for zero-shot image classification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4525–4534, 2017. 2

[48] Varun Khurana, Yaman Kumar Singla, Nora Hollenstein, Rajesh Kumar, and Balaji Krishnamurthy. Synthesizing human gaze feedback for improved nlp performance. *arXiv preprint arXiv:2302.05721*, 2023. 2

[49] Hanjoo Kim, Minkyu Kim, Dongjoo Seo, Jinwoong Kim, Heungseok Park, Soeun Park, Hyunwoo Jo, KyungHyun Kim, Youngil Yang, Youngkwan Kim, et al. Nsml: Meet the mlaas platform with a real-world case study. *arXiv preprint arXiv:1810.09957*, 2018. 9

[50] Nam Wook Kim, Zoya Bylinskii, Michelle A. Borkin, Krzysztof Z. Gajos, Aude Oliva, Fredo Durand, and Hanspeter Pfister. Bubbleview: An interface for crowdsourcing image importance maps and tracking visual attention. *ACM Trans. Comput.-Hum. Interact.*, 24(5), 2017. 2

[51] Nam Wook Kim, Zoya Bylinskii, Michelle A. Borkin, Aude Oliva, Krzysztof Z. Gajos, and Hanspeter Pfister. A crowdsourced alternative to eye-tracking for visualization understanding. In *33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, page 1349–1354, 2015. 2

[52] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *37th International Conference on Machine Learning*, pages 5338–5348, 2020. 2

[53] John Lambert, Ozan Sener, and Silvio Savarese. Deep learning under privileged information using heteroscedastic dropout. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8886–8895, 2018. 2

[54] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature Communications*, 10(1):1–8, 2019. 2, 3

[55] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014. 3, 4, 5, 6, 14

[56] Drew Linsley, Sven Eberhardt, Tarun Sharma, Pankaj Gupta, and Thomas Serre. What are the visual features underlying human versus machine vision? In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2706–2714, 2017. 2

[57] Drew Linsley, Dan Shiebler, Sven Eberhardt, and Thomas Serre. Learning what and where to attend with humans in the loop. In *International Conference on Learning Representations*, 2019. 2

[58] Chengjiang Long, Gang Hua, and Ashish Kapoor. Active visual recognition with expertise estimation in crowdsourcing. In *IEEE International Conference on Computer Vision*, pages 3000–3007, 2013. 2

[59] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. 6

[60] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 6

[61] Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10437–10446, 2020. 6

[62] Zihang Meng, Licheng Yu, Ning Zhang, Tamara L Berg, Babak Damavandi, Vikas Singh, and Amy Bearman. Connecting what to say with where to look by modeling human attention traces. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12679–12688, 2021. 2

[63] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995. 3

[64] Fintan Nagle and Nilli Lavie. Predicting human complexity perception of real-world scenes. *Royal Society open science*, 7(5):191487, 2020. 2

[65] Frederik Pahde, Maximilian Dreyer, Wojciech Samek, and Sebastian Lapuschkin. Reveal to revise: An explainable ai life cycle for iterative bias correction of deep models. *arXiv preprint arXiv:2303.12641*, 2023. 3

[66] Dim P Papadopoulos, Alasdair DF Clarke, Frank Keller, and Vittorio Ferrari. Training object class detectors from eye tracking data. In *European Conference on Computer Vision*, pages 361–376. Springer, 2014. 2

[67] Joshua C Peterson, Ruairidh M Battleday, Thomas L Griffiths, and Olga Russakovsky. Human uncertainty makes classification more robust. In *IEEE/CVF International Conference on Computer Vision*, pages 9617–9626, 2019. 2

[68] Suzanne Petryk, Lisa Dunlap, Keyan Nasseri, Joseph Gonzalez, Trevor Darrell, and Anna Rohrbach. On guiding visual attention with language specification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18092–18102, 2022. 3

[69] Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. Connecting vision and language with localized narratives. In *European Conference on Computer Vision*, 2020. 2

[70] Sukrut Rao, Moritz Böhle, Amin Parchami-Araghi, and Bernt Schiele. Using explanations to guide models. *arXiv preprint arXiv:2303.11932*, 2023. 3

[71] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, 2019. 2, 3, 6, 15

[72] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? *arXiv preprint arXiv:1902.10811*, 2019. 3, 4, 14

[73] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28, 2015. 8

[74] Zhongzheng Ren, Zhiding Yu, Xiaodong Yang, Ming-Yu Liu, Alexander G Schwing, and Jan Kautz. Ufo 2: A unified framework towards omni-supervised object detection. In *European Conference on Computer Vision*, pages 288–313. Springer, 2020. 2

[75] Yao Rong, Wenjia Xu, Zeynep Akata, and Enkelejda Kasneci. Human attention in fine-grained classification. *arXiv preprint arXiv:2111.01628*, 2021. 2

[76] Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. In *Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 2662–2670, 2017. 3, 6

[77] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 2, 3, 14

[78] Jeffrey M Rzeszotarski and Aniket Kittur. Instrumenting the crowd: using implicit behavioral measures to predict task performance. In *24th Annual ACM symposium on User Interface Software and Technology*, pages 13–22, 2011. 2

[79] Christoph Sager, Christian Janiesch, and Patrick Zschech. A survey of image labelling for computer vision applications. *Journal of Business Analytics*, 4(2):91–110, 2021. 3

[80] Walter J Scheirer, Samuel E Anthony, Ken Nakayama, and David D Cox. Perceptual annotation: Measuring human vision to improve computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1679–1686, 2014. 2

[81] Vaishaal Shankar, Rebecca Roelofs, Horia Mania, Alex Fang, Benjamin Recht, and Ludwig Schmidt. Evaluating machine accuracy on imagenet. In *International Conference on Machine Learning*, pages 8634–8644. PMLR, 2020. 3

[82] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019. 5

[83] Viktoriia Sharmanska, Daniel Hernández-Lobato, Jose Miguel Hernandez-Lobato, and Novi Quadrianto. Ambiguity helps: Classification with disagreements in crowdsourced annotations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2194–2202, 2016. 2

[84] Viktoriia Sharmanska, Novi Quadrianto, and Christoph H Lampert. Learning to rank using privileged information. In *International Conference on Computer Vision*, pages 825–832, 2013. 2

[85] Viktoriia Sharmanska, Novi Quadrianto, and Christoph H Lampert. Learning to transfer privileged information. *arXiv preprint arXiv:1410.0389*, 2014. 2

[86] Victor S Sheng, Foster Provost, and Panagiotis G Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 614–622, 2008. 2

[87] Rakshith Shetty, Bernt Schiele, and Mario Fritz. Not using the car to see the sidewalk–quantifying and controlling the effects of context in classification and segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8218–8226, 2019. 2, 3, 5, 8

[88] Becks Simpson, Francis Dutil, Yoshua Bengio, and Joseph Paul Cohen. Gradmask: Reduce overfitting by regularizing saliency. *arXiv preprint arXiv:1904.07478*, 2019. 3, 6

[89] Yuru Song, Zan Lou, Shan You, Erkun Yang, Fei Wang, Chen Qian, Changshui Zhang, and Xiaogang Wang. Learning with privileged tasks. In *IEEE/CVF International Conference on Computer Vision*, pages 10685–10694, 2021. 2

[90] Ekta Sood, Simon Tannert, Philipp Müller, and Andreas Bulling. Improving natural language processing tasks with human gaze-guided neural attention. *Advances in Neural Information Processing Systems*, 33:6327–6341, 2020. 2

[91] Megha Srivastava, Tatsunori Hashimoto, and Percy Liang. Robustness to spurious correlations via human annotations. In *International Conference on Machine Learning*, 2020. 2

[92] Yusuke Sugano and Andreas Bulling. Seeing with humans: Gaze-assisted neural image captioning. *arXiv preprint arXiv:1608.05203*, 2016. 2

[93] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016. 19

[94] Ece Takmaz, Sandro Pezzelle, Lisa Beinborn, and Raquel Fernández. Generating image descriptions via sequential cross-modal alignment guided by human gaze. *arXiv preprint arXiv:2011.04592*, 2020. 2

[95] Tian Tian and Jun Zhu. Max-margin majority voting for learning from crowds. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, 2015. 2

[96] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, 2021. 6, 8, 19, 20, 21

[97] Radu Tudor Ionescu, Bogdan Alexe, Marius Leordeanu, Marius Popescu, Dim P Papadopoulos, and Vittorio Ferrari. How hard can it be? estimating the difficulty of visual search in an image. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2157–2166, 2016. 2, 5

[98] Wil MP Van der Aalst, Boudewijn F Van Dongen, Joachim Herbst, Laura Maruster, Guido Schimm, and Anton JMM Weijters. Workflow mining: A survey of issues and approaches. *Data & knowledge engineering*, 47(2):237–267, 2003. 2

[99] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8769–8778, 2018. 3

[100] Vladimir Vapnik. *Estimation of dependences based on empirical data*. Springer Science & Business Media, 2006. 2

[101] Vladimir Vapnik. Learning with teacher: Learning using hidden information. In *Proc. International Joint Conference on Neural Networks*, volume 2009, 2009. 2

[102] Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5):544–557, 2009. 2

[103] Luis von Ahn. Human computation. In *4th International Conference on Knowledge Capture*, page 5–6. Association for Computing Machinery, 2007. 1

[104] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. *Tech report*, 2011. 3

[105] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, 2019. 6

[106] Philippe Weinzaepfel and Grégory Rogez. Mimetics: Towards understanding human actions out of context. *International Journal of Computer Vision*, 129(5), 2021. 3

[107] Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. *ArXiv preprint arXiv:2006.09994*, 2020. 5, 6, 7

[108] Heng Yang and Ioannis Patras. Privileged information-based conditional regression forest for facial feature detection. In *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*, pages 1–6, 2013. 2

[109] Hao Yang, Joey Tianyi Zhou, Jianfei Cai, and Yew Soon Ong. Miml-fcn+: Multi-instance multi-label learning via fully convolutional networks with privileged information. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1577–1585, 2017. 2

[110] Huihan Yao, Ying Chen, Qinyuan Ye, Xisen Jin, and Xiang Ren. Refining language models with compositional explanations. *Advances in Neural Information Processing Systems*, 34, 2021. 2

[111] Kiwon Yun, Yifan Peng, Dimitris Samaras, Gregory J Zelinsky, and Tamara L Berg. Studying relationships between human gaze, description, and computer vision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 739–746, 2013. 2

[112] 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 *IEEE/CVF International Conference on Computer Vision*, 2019. 6, 19, 21

[113] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3712–3722, 2018. 6

[114] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 6, 19, 21

[115] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016. 8

[116] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2017. 3

Appendix

We include additional information in the Appendix. In §A, you can download ImageNet-AB and COCO-AB datasets. In §B, you will find the directories for front-end code for ImageNet and COCO annotation tools. In §C, we present details for our crowdsourcing-based ImageNet and COCO re-annotations. In §D, we present extensive lists of byproducts from ImageNet-AB and COCO-AB. In §E, we present further statistics and interesting features of the annotation byproducts in ImageNet-AB and COCO-AB. In §F, we include additional experimental details and results that supplement the main-paper results.

A. Download links

- ImageNet-AB (Click to start downloading; 529MB)
- COCO-AB (Click to start downloading; 380MB)

B. Front-end code for ImageNet and COCO

Please find the codebase for ImageNet and COCO annotation tools in the root directory:

- ImageNet: github.com/naver-ai/imagenet-annotation-tool
- COCO: github.com/naver-ai/coco-annotation-tool

They are replications of respective original annotation tools: [77, 72] for ImageNet and [55] for COCO.

C. Annotation and crowdsourcing details

C.1. ImageNet

We provide further details on the crowdsourced ImageNet annotation. We hired Amazon Mechanical Turk (MTurk) workers from the US region, as the task is described in English. The minimal human intelligence task (HIT) approval rate for the task qualification was set at 90% to ensure a minimal quality for the task.

Each HIT contains 10 pages of the annotation task, each with 48 candidate images. Upon completion, the annotators are paid 1.5 USD per HIT. It is difficult to convert this amount to an exact hourly wage due to the high variance and noise in the measured time to complete each HIT. A rough conversion is possible through the median HIT, which took 9.0 minutes to complete. This yields an hourly wage of 10.0 USD, well above the US federal minimum hourly wage of 7.25 USD [3].

When the submitted work shows clear signs of gross negligence and irresponsibility, we reject the HIT. Specifically, we reject a HIT if:

- the recall rate, defined as the proportion of selected images I_c^{select} among the original ImageNet subset I_c^{in}, is lower than 0.333; or
- the total number of selections I_c^{select} among 480 candidates is lower than 30 (there are $480 \times 0.75 = 360$ samples from ImageNet I_c^{in} on average); or
- the annotator has not completed at least 9 out of the 10 pages of tasks; or
- the annotation is not found in our database AND the secret hash code for confirming their completion is incorrect.

Among 14,681 HITs completed, 1,145 (7.8%) have been rejected. Collectively, we have paid $20,304$ USD $= 13,536$ approved HITs $\times$ 1.5 USD / HIT to the MTurk annotators. An additional 20% fee is paid to Amazon ($4,060.8$ USD). The entire procedure took place between 18 December 2021 and 31 December 2021.

Annotation interface. We have tried nudging the annotators to click more frequently on the foreground objects by changing the cursor shape to a red circle and instructing them to "click on the object of interest" while selecting the images. According to our pilot study, this increases the chance of annotators clicking on the object of interest from 70.7% to 91.7% (p-value <0.0005), while not increasing the annotation time meaningfully: 2.02 to 2.09 minutes per page (p-value 0.456).

C.2. COCO

For COCO, we follow the ImageNet annotation setup in §C.1 for the worker region and worker qualification.

Each annotation page contains a single image to be annotated. We collate 20 pages into a single human intelligence task (HIT). That results in $82,783$ images $\times \frac{1 \text{ HIT}}{20 \text{ images}} = 4,140$ HITs. The compensation for each HIT is 2.0 USD. The median HIT has been completed in 12.1 minutes. This leads to the hourly wage of 9.92 USD, which is above the US Federal minimum wage of 7.25 USD [3].

We reject HITs based on the following criteria

- the recall rate, defined as the proportion of retrieved classes among the existing classes, is lower on average than 0.333; or
- the accuracy of icon location, defined as the ratio of icons placed on the ground-truth class segmentation mask, is lower than 0.75; or
- the annotator has not completed at least 16 out of the 20 pages of tasks; or
- the annotation is not found in our database AND the secret hash code for confirming their completion is incorrect.

By continuously re-posting rejected HITs, we have acquired the necessary annotation and byproducts on 4140

```
  "imageID": "n01440764/n01440764_105",
  "originalImageHeight": 375,
  "originalImageWidth": 500,
  "selected": true,                    Original Annotation
  "imageHeight": 243,
  "imageWidth": 243,
  "imagePosition": {"x": 857, "y": 1976},
  "hoveredRecord": [
    {"action": "enter", "time": 1641425051},
    {"action": "leave", "time": 1641425319}
  ],
  "selectedRecord": [
    {"x": 0.540, "y": 0.473, "time": 1641425052}
  ],
  "mouseTracking": [
    {"x": 0.003, "y": 0.629, "time": 1641425051},
    {"x": 0.441, "y": 0.600, "time": 1641425052}
  ],
  "worker_id": "47DBDD543E",
  "assignment_id": "3AMYWKA6YLE80HK9QYYHI2YEL2YO6L",
  "page_idx": 3                    Annotation Byproducts
```


Figure A: **Annotation byproducts from ImageNet.** Worker ID has been anonymised via non-reversible hashing. Extended version of Figure 4.

HITs. Along the way, we have rejected 365 HITs, giving us a rejection rate 8.8%. Collectively, we have paid $8,280$ USD $= 4,140$ approved HITs $\times$ 2 USD / HIT to 662 MTurk annotators. An additional 20% fee is paid to Amazon (1656 USD). The annotation took place between 9 January 2022 and 12 January 2022.

D. Byproducts details

D.1. ImageNet-AB

We explain the details of ImageNet-AB, the ImageNet1k training set enriched with annotation byproducts. Annotators use input devices to interact with different components in the annotation interface. This results in a history of interactions per input signal per front-end component. On ImageNet, annotators interact with each image (component) on each page with two types of input signals: mouse movements and mouse clicks (Figure 3). We show the full list of annotation byproducts in Figure A. This results in the time series of mouse movements (`mouseTracking`) and mouse clicks (`selectedRecord`) for every image. We separately record whether the image is finally selected by the annotator in the `selected` field. It is `true` when the length of `selectedRecord` is an odd number.

In our work, we only demonstrate the usage of additional `selectedRecord` as a proxy to the object localisation information and show that this alone greatly enhances the models' robustness. However, there exist other byproducts that may further improve the trained models. We introduce them below and hope that future researches find ways to maximally exploit those additional signals.

We record sufficient yet compact information to reproduce the annotation page: x-y coordinates (`imagePosition`) and the width and height (`imageWidth` and `imageHeight`) of each image in the annotation interface. This information can be useful because the mouse movement pattern is highly entangled with the page layout. For example, annotators are likely to minimise mouse movement by following a serpentine sequence.

We record other annotation metadata for each image, such as the worker identifier (`worker_id`), the identifier for the human intelligence task (HIT) that contains this image (`assignment_id`), and the page number within the HIT (`page_idx`). We have anonymised the worker identifier with a non-reversible hashing function. Those metadata provide information for grouping the annotation instances with increasing specificity: {annotations on the same page} $\subset$ {annotations from the same HIT} $\subset$ {annotations by the same worker}. Such information may be helpful for identifying and factoring out group-specific idiosyncrasies. For example, worker `ABC` may always click near the centre of an image; we may then decide not to use her clicks as a reliable estimate of object locations. Or we may find that the HIT `DEF` was done in such a rush; we would then reduce the weight for the set of annotations belonging to `DEF`.

Statistics. There are 1,281,167 ImageNet1K training images I^{imagenet}. There were two annotation rounds. In the first round, human intelligence tasks (HITs) containing all 1,281,167 original images are shown to the annotators. They have re-selected 71.8% of them. This confirms the observation of [71] that 71% of the validation set samples were re-selected in their setting. The remaining 28.2% of I^{imagenet} are re-packaged into a second batch of HITs and presented to the annotators. They have additionally re-selected 14.9% of I^{imagenet}, resulting in the final 1,110,786 (86.7%) ImageNet1K training images that are re-selected. Those selected images now come with rich annotation byproducts, such as the time-series of mouse traces and clicks. However, annotation byproducts are available even for images that are not finally selected; they are recorded even for images that annotators cancel the selection or simply hover the cursor over. As a result, 1,272,225 (99.3%) of the ImageNet1K training set have any form of annotation byproduct available.

D.2. COCO-AB

We explain the details of COCO-AB, the COCO 2014 training set enriched with annotation byproducts. COCO interface (Figure 5) has two main components: (1) the image on which the class icons are placed and (2) the class browsing tool showing the class icons. The annotation byproducts come from those two sources. See Figure B for the full list of annotation byproducts.

The `actionHistories` field describes the actions performed with the mouse cursor on the image. `actionHistories` list the sequence of actions with possible types `add`, `move`, and `remove` and the corresponding

```
"image_id": 459214,
"originalImageHeight": 428,
"originalImageWidth": 640,
"categories": ["car", "bicycle"], Original Annotation
"imageHeight": 450,
"imageWidth": 450,
"timeSpent": 22283,
"actionHistories": [
  {"actionType": "add",
   "iconType": "car",
   "pointTo": {"x": 0.583, "y": 0.588},
   "timeAt": 16686},
  {"actionType": "add",
   "iconType": "bicycle",
   "pointTo": {"x": 0.592, "y": 0.639},
   "timeAt": 16723}
],
"categoryHistories": [
  {"categoryIndex": 1,
   "categoryName": "Animal",
   "timeAt": 10815,
   "usingKeyboard": false},
  {"categoryIndex": 10,
   "categoryName": "IndoorObjects",
   "timeAt": 19415,
   "usingKeyboard": false}
],
"mouseTracking": [
  {"x": 0.679, "y": 0.862, "timeAt": 15725},
  {"x": 0.717, "y": 0.825, "timeAt": 15731}
],
"worker_id": "00AA3B5E80",
"assignment_id": "3AMYWKA6YLE80HK9QYYHI2YEL2YO6L",
"page_idx": 8                    Annotation Byproducts
```


Figure B: **Annotation byproducts from COCO.** Worker ID has been anonymised via non-reversible hashing. Extended version of Figure 6.

location and time. We also record the object class of the icon. The `mouseTracking` field records the movement of the mouse cursor over the image.

Interactions with the class browsing tool leave a time series of superclasses that the annotator refers to. They are stored in the field `categoryHistories`. We also allow interactions based on keyboard (left and right arrows); the use of keyboard is indicated in `usingKeyboard`.

We record the total time spent for the annotation (`timeSpent`). To provide the context of the annotation work, we have stored the page number (`page_idx`), the identifier for the HIT package (`assignment_id`), and the anonymised identifier for the annotator (`worker_id`).

In this work, we only use the last `add` action in the `actionHistories` field for each object class to additionally supervise the model to be aware of the actual location of the object in the image. However, the recordings of other interaction histories may be used in future work as additional sources that further improve the trained models.

Statistics. Annotators have reannotated 82,765 (99.98%) of 82,783 training images from the COCO 2014 training set. For those images, we have recorded the annotation byproducts. We found that each HIT recalls 61.9% of the list of classes per image, with the standard deviation $\pm$0.118%p. The average localisation accuracy for icon placement is 92.3% where the standard deviation is $\pm$0.057%p.

E. Analysis of annotation byproducts

E.1. ImageNet

We analyse the annotation byproducts in more detail. In particular, we measure the informativeness of mouse clicks and traces for the location of objects in an image. All analyses involving the "ground-truth (GT) bounding boxes" is performed on the 42% of the ImageNet1K training set annotated with instance-wise bounding boxes.

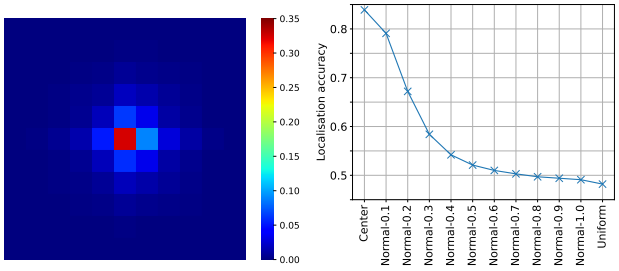


Figure C: **ImageNet GT-box statistics. Left**: distribution of GT box centres on ImageNet1K training set images. **Right**: localisation accuracy of random clicks $N((\frac{H}{2}, \frac{W}{2}), \sigma^2)$. We interpolate between centre-always click ($\sigma = 0$) and uniform random click ($\sigma = \infty$).

GT bounding boxes on ImageNet. ImageNet is a highly object-centric dataset. This is reconfirmed by the distribution of the centre of the GT boxes in Figure C (left). More than 30% of the box centres are located in the 0.82% area at the centre of the images.

We measure the localisation accuracy of random image-agnostic clicks in Figure C (right). We experimented with the random click distribution $N((\frac{H}{2}, \frac{W}{2}), \sigma^2)$ where $\sigma \in [0, \infty]$ interpolates between the click-always-at-the-centre strategy ($\sigma = 0$) and the uniform random click ($\sigma = \infty$). We observe that clicking at the image centre yields 83.9% localisation accuracy, actually greater than the localisation accuracy of clicks 82.9%. Despite a lower overall accuracy, we will see later in the current section that the annotators' clicks contain much richer information about the variation of object locations than simple centre clicks.

As σ increases, the localisation accuracy drops and reaches 48.2% when clicks are uniformly random $\sigma = \infty$. The 48.2% value can be interpreted as the average bounding box area in each image. The relatively high average area of the objects again signifies the object-centric nature of the ImageNet dataset.

Informativeness of clicks. We examine whether the clicks contain information about the variation of object locations. The analysis is not as simple as measuring the overall localisation accuracy, since the dataset is highly object-centric: we have seen above that centre clicks already give 83.9% localisation accuracy, greater than the localisation accuracy of

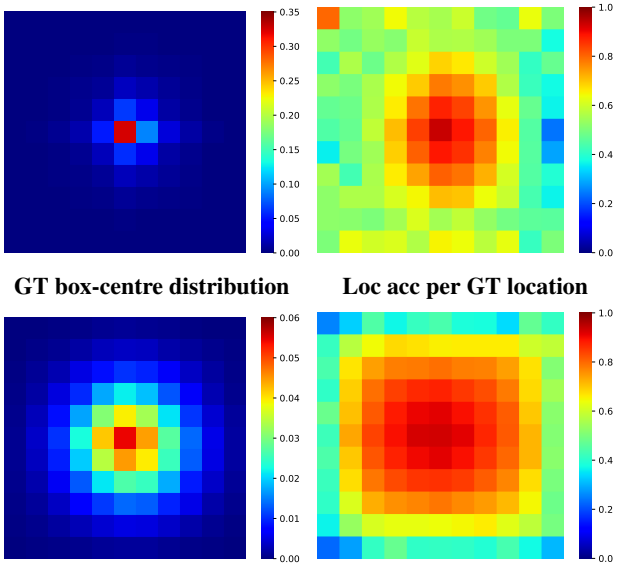


Figure D: **Statistics of clicks. Left column**: distribution of GT box centres and clicks in ImageNet1K images. **Right column**: localisation accuracy of clicks at each GT box centre location and click location.

clicks 82.9%. The majority of information about the object location is contained in 16.1% of the samples where a simple centre-click strategy cannot guarantee a correct localisation. In this subset of images where objects are not at the centre, the localisation accuracy of clicks is 56.5%. This implies great information content, as simple centre clicks will give 0% accuracy on this subset.

To further break down the localisation accuracy based on the location of objects and click locations, we plot the location-wise click accuracy in Figure D (right column). For reference, we also plot the distribution of GT box centres and clicks in the left column. We observe that the localisation accuracy at each GT box location and the click location remain $> 40\%$, except at the outermost image borders. This confirms the overall informativeness of clicks for the object locations, despite the severe bias towards the image centre in the dataset.

Informativeness of mouse traces. Annotation byproducts include not only clicks but the full history of mouse traces over each image. We measure the localisation accuracy of the mouse traces between entering the image and click. The results are reported in Figure E. Last few mouse trace records before click (Last N) show a mild drop in accuracy (from 82.9% to $\sim 65\%$ at 8 traces before click); therefore, the last few points before click may give useful localisation information. The trace and time quantile results show that the localisation accuracy is very low when the mouse enters an image (39.3%). The accuracy increases up

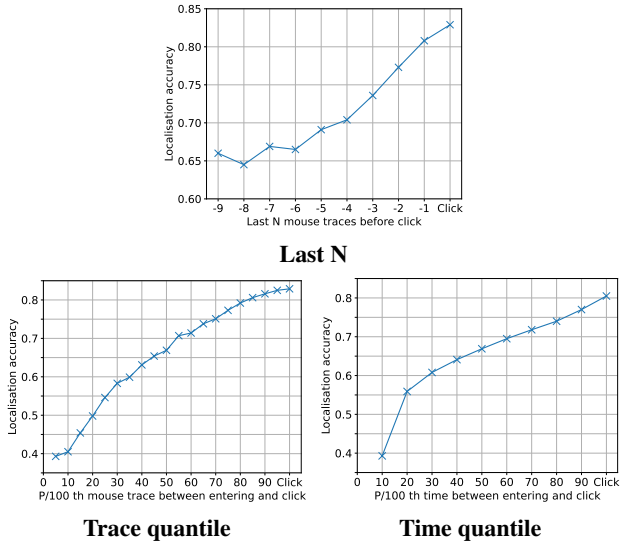


Figure E: **Statistics for mouse traces before click. Last N**: last N mouse traces before click. **Trace quantile**: division of each mouse trace from the "entering image" event to the "click" event in the equal number of mouse track records. **Time quantile**: same as trace quantile, except that bins are groups by the time.

to the point when the user clicks (82.9%). We observe that the last 10% of the mouse traces (both for trace and time quantile) are still fairly precise with accuracy $> 80\%$. The above observations imply the possibility that one may also utilise a few mouse trace records before the click event to obtain a weak localisation supervision based on scribbles [6].

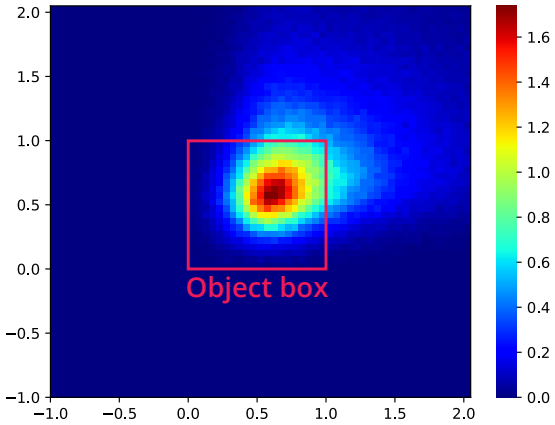


Figure F: **Click histogram relative to GT box on ImageNet.** Distribution of click positions normalised against the GT object box frame at $[0, 1] \times [0, 1]$.

Click are systematically biased to the top-right corner. Figure F shows the distribution of clicks relative to the GT object boxes. We observe that the mode of the distribution is close to the centre, but slightly biased to the upper-right corner. The tail of the distribution is more drastically biased

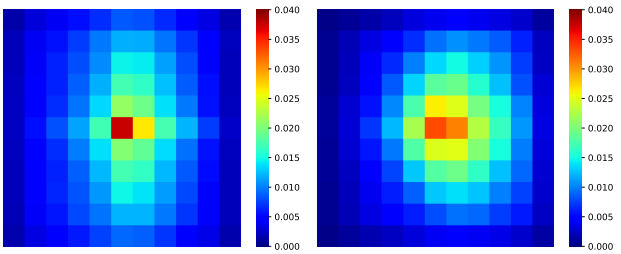


Figure G: **Statistics of icon placement.** Statistics for the location of objects and the final icon placements.

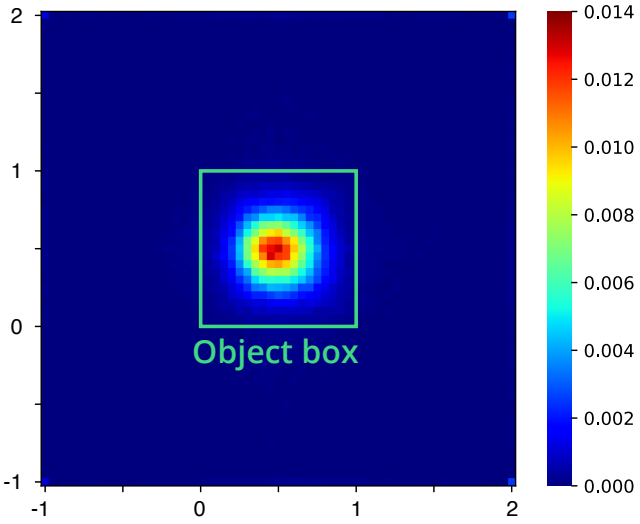


Figure H: **Icon histogram relative to the GT box on COCO.** Distribution of final "add" positions normalised against the GT object box frame at $[0, 1] \times [0, 1]$.

towards the top-right corner, almost forming a comet-like shape. We conjecture that browsing through rows of images makes annotators enter an image through the top side and leave it through the right side. And this leaves such a systematic error around the actual location of the objects. Given the systematic bias, it would be an interesting future research direction to either post-hoc calibrate click locations or nudge annotators to reduce the top-right-corner bias for better object localisation.

E.2. COCO

Distribution of objects in COCO. COCO is designed to contain multiple objects in the same image. We verify this by computing the histogram of the centres for COCO bounding boxes. Figure G (left) shows the distribution. Compared to ImageNet (Figure C left), we observe more diffused box centres in COCO. As a result, we observe more diffused object centres for the COCO objects within an image. There are less than 4% instances in the centre of the image; the ratio was greater than 30% for ImageNet.

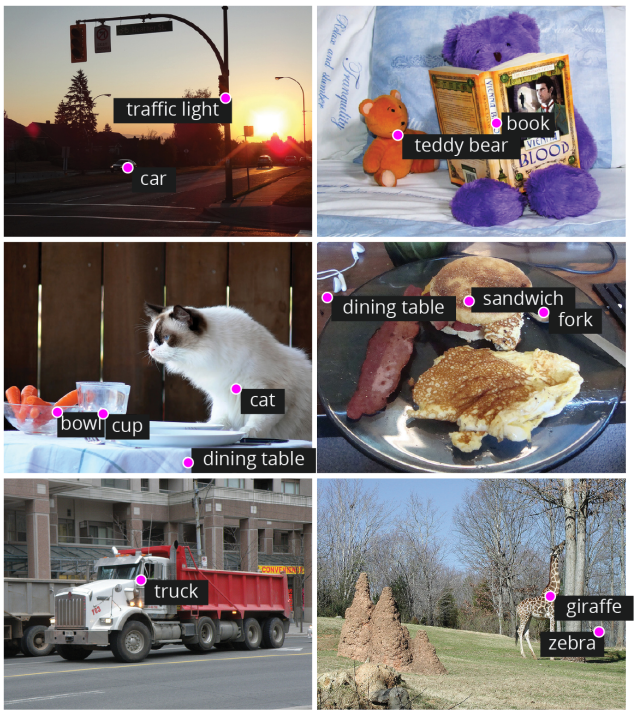


Figure I: **COCO final icon locations**. We visualise random training images; **points** are the final location of the `add` action for each category in `actionHistories`.

```
154320 (94%)  add
  4128 ( 3%)  add-move
  2778 ( 2%)  add-remove-add
   344 ( 0%)  add-move-move
   271 ( 0%)  add-remove-add-remove-add
   191 ( 0%)  add-move-remove-add
   114 ( 0%)  add-remove-add-move
    67 ( 0%)  add-remove-add-remove-add-remove-add
    37 ( 0%)  add-move-remove-add-move
    29 ( 0%)  add-move-move-remove-add
    29 ( 0%)  add-move-move-move
    27 ( 0%)  add-move-remove-add-remove-add
    19 ( 0%)  add-remove-add-remove-add-remove-add-remove-add
    17 ( 0%)  add-remove-add-remove-add-move
    12 ( 0%)  add-remove-add-move-remove-add
    11 ( 0%)  add-move-move-move-move
```

Figure J: **Histogram of action sequences on COCO.** Only showing action sequences with > 10 occurrences.

Icon placements. Example locations of icon placements are shown in Figure I. The distribution of icon placement locations on COCO images is shown in Figure G (right). We observe a distribution that is similar to the box-centre distribution, confirming the fairly precise icon placement accuracy of 92.3% (§D.2). We also measure the systematic bias in icon placement with respect to ground-truth bounding boxes in Figure H. We observe no visible bias. This is in stark contrast to the ImageNet click locations in Figure F. We hypothesise that the tagging interface lets annotators be more focused and be careful with the relative location of the icons with respect to the object regions.

Action sequences in COCO annotations. Annotators can perform three types of actions with the icons: `add`, `move`, and `remove`. In Figure J, we show the histogram of the action sequences for icons that are eventually placed in the images. The most frequent action sequence is a singleton `add` with 94% frequency. The next common sequence is `add-move` with 3% frequency: the annotator corrects the position once. The third most frequent sequence is `add-remove-add` with 2% frequency: the annotator removes the placed icon and then adds it back. This could indicate the annotator's lack of confidence in either the position of the object or the existence of the object. There are other interesting behaviours. For example, 19 action sequences repeat the addition and removal: `(add-remove)*4-add`. We are not sure if this behaviour is due to the annotator's uncertainty or is due to no particular reason (for example, just for fun). In fact, the longest action sequence was `add-remove-add-move-(remove-add)*7-move-move-(remove-add)*2` (24 actions).

Recall by category and object sizes. We study whether the size of objects contributes to the successful annotation of the object. Figure K shows the scatter plot for class-wise recall versus class-wise average size. Class-wise recall measures the chance that an instance of the class in an image is annotated via icon placement. Class-wise sizes are measured by binning the object box by bins $[0, .2^2, .4^2, .6^2, .8^2, 1]$. We observe a linear correlation between the object sizes and the recall. This indicates that larger object categories are more likely to be annotated than smaller ones. There are interesting exceptions. For example, sports equipment such as "tennis racket", "skateboard", "baseball racket", "frisbee" and "sports ball" tends to be annotated successfully compared to their small size. We expect this to be related to the saliency of objects. Sports equipment is likely designed to attract human attention or humans are trained to detect such objects well. In the opposite regime, we find furniture such as "bed" and "dining table" is less frequently annotated compared to its size. Again, we believe its relative saliency results in low recall. We tend to perceive such furniture more as a background object that is easy to be overlooked in a scene.

F. Additional experimental details

Training details. For the ImageNet experiments, we use all the default training hyperparameters provided in the DeiT [96] codebase[2] including training epochs 300 with warmup epochs 5, batch size 1024, learning rate 5e-4$\times \frac{\text{batchsize}}{512}$, weight decay 0.05. In addition, we use the default hyperparameters for data augmentations and regularizations – RandAug [16] 9/0.5 (*i.e.* rand-m9-mstd0.5-inc1), Label smoothing [93] 0.1, Stochastic Depth 0.1 with the linear decay of death rate [44], and Random Erasing [41, 19] 0.25; Mixup [114] and Cutmix [112] with the probabilities 0.8 and 1.0, respectively with switching probability 0.5, and the repeated augmentation [42] with 3 repetitions. We train the models with the image size of 224×224 and the test crop ratio of 0.875 based on the basic ImageNet training strategy – RandomResizedCrop, RandomFlip, and ColorJitter following the standard protocol [35, 21, 96]. All the models are trained with the multi-task objective using λ=10.

For the COCO experiments, there is no standard configuration for the image classification task, so we search for hyperparameter sets for convergence of the baseline networks. As a result, we set training epochs to 100 (5 for warmup epochs), batch sizes to 128, image size to 224×224, learning rate to 2e−5, and weight decay to 0.01. We use the standard data augmentation of the aforementioned basic ImageNet training strategy for all models. In addition to this, we set the minimum range of RandomResizedCrop to 0.1, and use Random Erasing [41, 19] with 0.5. Specifically, we only use We use the AdamP [40] optimizer for training all backbone networks. For multi-task learning, we observe that small λ works well with the small backbone network, and large λ is more effective for larger backbone networks. Specifically, we used λ=5 for ResNet18 and ViT-Ti. We used λ=50 for ResNet50, ResNet152, ViT-S, and ViT-B. Figure L shows that, across all λ, `LUAB` performs generally better than the models trained with Random points (Rand) or only with task supervision (*i.e.* λ=0).

Visualisation of the predicted points. We visualise the points predicted by our `LUAB`-trained models with the annotation byproducts. Figure M and N show the points predicted by our ViT-B in random ImageNet validation images and by our ResNet50 in random COCO validation images, respectively. We observe the predicted points are aligned with the ground-truth object locations.

Using annotation byproducts for data-efficient learning. Table C shows ViT-Ti performances after training with varying amounts of training data. The result shows that we may use 95% of ImageNet training data without decreasing the performance when annotation byproducts are utilised.

Using annotation byproducts to pool features. In the main paper, we have introduced a multi-task learning approach with the point-regression objective for the annotation byproducts. Here, we show another possibility to use the annotation byproducts. We use them as ground-truth attention for a weighted pooling for a convolutional neural network. We design a network architecture with a point-guided (*i.e.* attentive) pooling layer that amplifies the features corresponding to the point coordinates. The experimental result in Ta-

[2] https://github.com/facebookresearch/deit

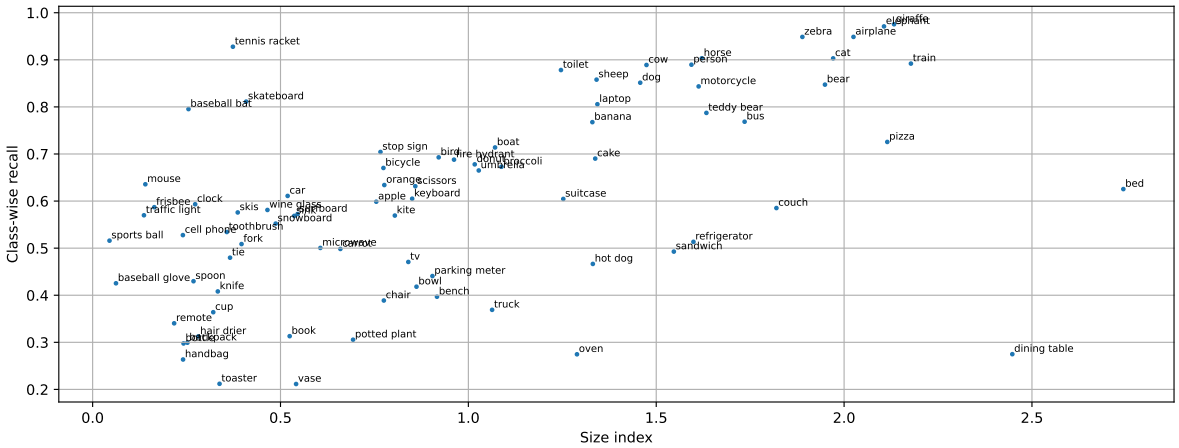


Figure K: **Recall versus size for each COCO category.**

Model	Params	IN-1k↑	IN-V2↑	IN-Real↑	IN-A↑	IN-C↑	IN-O↑	Sketch↑	IN-R↑	Cocc↑	ObjNet↑	SI-size↑	SI-loc↑	SI-rot↑	BGC-gap↓	BGC-acc↑
R18	11.7M	71.8	59.7	74.4	**1.9**	37.1	52.6	**21.9**	33.8	42.7	21.8	47.5	22.2	31.9	8.6	**22.4**
+LUAB	11.7M	**72.0**	**59.9**	**74.5**	1.8	**37.8**	**52.6**	21.7	**33.8**	**43.6**	**22.0**	**47.6**	**23.5**	**32.2**	**7.4**	20.1
R50	25.6M	77.2	65.4	78.2	4.6	39.8	**57.5**	**25.4**	37.2	53.9	27.7	54.2	31.6	39.3	**6.0**	28.8
+LUAB	25.6M	**77.4**	**65.8**	**78.2**	**5.4**	**44.1**	56.2	25.1	**37.6**	**54.3**	**27.7**	**54.7**	**31.7**	**40.2**	6.4	**29.2**

Table A: **An alternative baseline of using annotation byproducts.** We report the performance of the models using annotation byproducts as guidance of feature pooling location at training. The performance improvements here show that this method can also become a potential approach for using annotation byproducts to improve the robustness and localization abilities. A more sophisticated method upon this baseline would improve the numbers more.

Model	Params	IN-1k↑	IN-V2↑	IN-Real↑	IN-A↑	IN-C↑	IN-O↑	Sketch↑	IN-R↑	Cocc↑	ObjNet↑	SI-size↑	SI-loc↑	SI-rot↑	BGC-gap↓	BGC-acc↑
ViT-Ti	5.7M	71.8	58.8	73.6	4.8	41.4	59.1	18.6	29.6	38.7	20.1	40.6	16.5	26.2	12.1	13.6
+LUAB	5.7M	**73.0**	**60.2**	**74.7**	**5.7**	**42.5**	**59.9**	**19.4**	**30.8**	**42.6**	**22.1**	**43.4**	**20.0**	**28.7**	**10.9**	**16.1**
ViT-S	22.1M	74.1	60.8	75.3	5.1	45.0	55.0	22.9	34.7	**47.0**	20.5	42.9	18.7	27.8	10.5	16.7
+LUAB	22.1M	**75.3**	**63.0**	**76.5**	**6.3**	**47.7**	**59.1**	**24.4**	**36.5**	46.6	**23.6**	**47.8**	**22.6**	**32.2**	**8.7**	**19.7**
ViT-B	86.6M	75.1	61.9	76.1	6.4	48.8	**56.8**	24.3	36.7	48.9	21.3	**47.6**	22.1	**31.9**	8.9	18.9
+LUAB	86.6M	**75.9**	**63.0**	**76.9**	**7.6**	**49.9**	56.5	**26.4**	**37.2**	**50.3**	**23.2**	47.4	**22.5**	31.7	**8.0**	**18.9**

Table B: **Performance of ImageNet-AB on ImageNet1K without sophisticated training recipes.** We extend the study in Table 1 by training ViTs [21, 96] with simpler training recipes. We note more significant improvements due to ImageNet-AB than shown in Table 1.

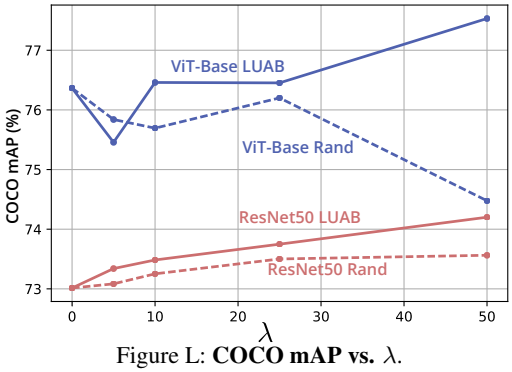
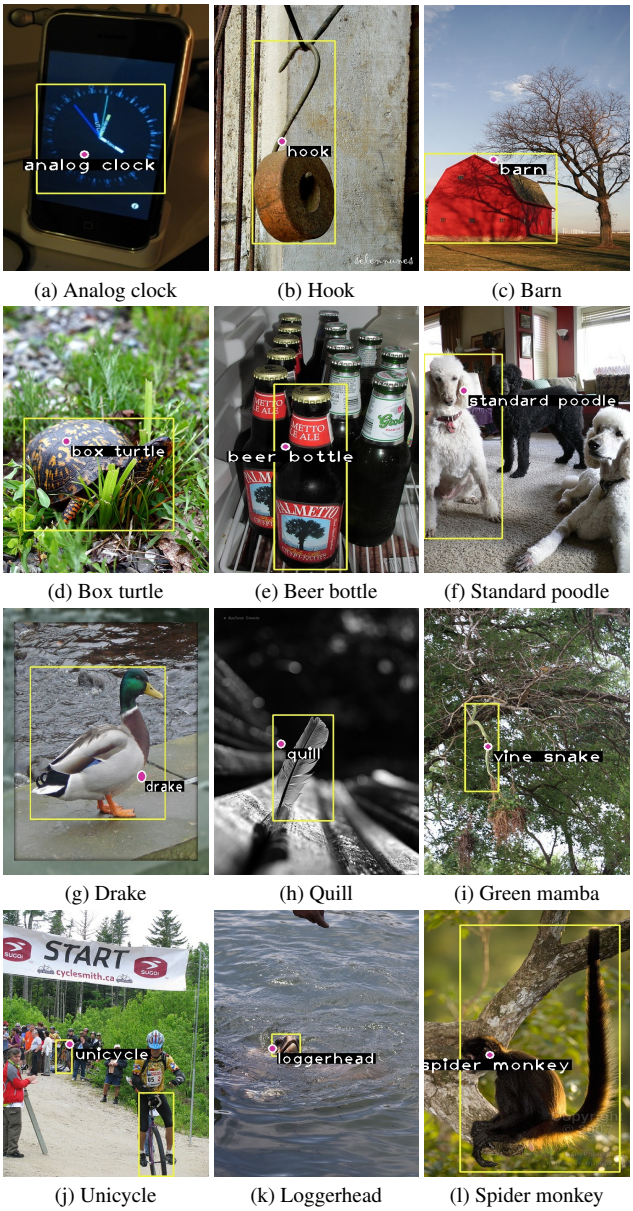


Figure L: **COCO mAP vs.** λ.

Training data	% Data used	ImageNet-1k acc (%)
ImageNet w/o AB	100%	72.8
ImageNet + AB	100%	**72.9**
	95%	**72.9**
	90%	72.4
	80%	71.7

Table C: **Data-efficient training with `LUAB`**. The availability of annotation byproducts (AB) let us use slightly less amount of training data (100%→95%).

ble A shows that this simple method (without any extensive hyperparameters tuning) improves the overall performance of ResNet18 and ResNet50. As for the multitask learning baseline, this attentive pooling approach improves classification performance, OOD generalisation, and resilience to spurious background correlations.



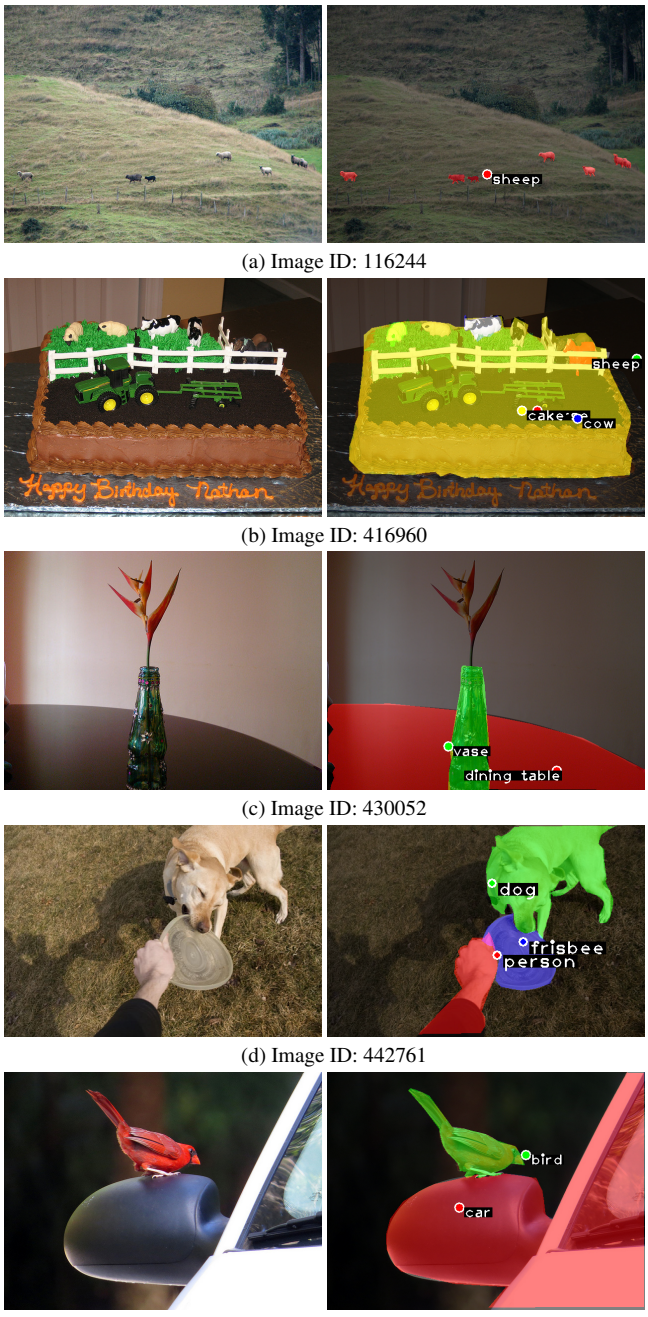
(a) Analog clock (b) Hook (c) Barn

(d) Box turtle (e) Beer bottle (f) Standard poodle

(g) Drake (h) Quill (i) Green mamba

(j) Unicycle (k) Loggerhead (l) Spider monkey

Figure M: **Model prediction visualisation (ImageNet)**. We visualise some validation images in ImageNet with the ground truth **boxes** and the predicted **points** by our model.



(a) Image ID: 116244

(b) Image ID: 416960

(c) Image ID: 430052

(d) Image ID: 442761

(e) Image ID: 217554

Figure N: **Model prediction visualisation (COCO)**. We visualise COCO validation images with the ground truth mask and predicted points by our model.

Impact of LUAB without strong augmentations. In the main paper, we have considered the backbones trained with strong augmentations (*e.g.* DeiT) to make the results more relevant to the state-of-the-art models. Here, we examine the impact of LUAB without such strong augmentations. We choose ViTs as the baseline models because they usually suffer from data deficiency [21, 96] and require stronger augmentations. We follow the training setup provided in original ViT [21]; we limit the strong data augmentation or regularisations previously used. Table B shows the performances without strong augmentations such as RandAug [16], Stochastic Depth [44], Random Erasing [41, 19], Mixup [114], Cutmix [112] in the DeiT training regime [96]. We use a training setup similar to the one in the ViT paper [21]: learning rate 1e-3 and weight decay 0.3. All the models are trained with the multi-task objective using λ=10 again. We observe that the performance improvements due to LUAB are much greater than those in Table 1. We conclude that the actual impact of annotation byproducts is greater when the performances are not optimised with the use of strong augmentations.