Appendix for Neglected Free Lunch – Learning Image Classifiers Using Annotation Byproducts

We include additional information in the Appendix. In §A, you can download ImageNet-AB and COCO-AB datasets and 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. Links

Our main repository is at:

- Neglected Free Lunch (GitHub)

Download datasets at:

- ImageNet-AB (HuggingFace)
- COCO-AB (HuggingFace)

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: [19, 15] for ImageNet and [14] for COCO.

B. Detailed comparison against previous work

We cluster the related work into two groups in Table A. Group A: Solving image classification with additional annotations (*e.g*. semantic segmentation) [18, 20, 5]. Group B: Solving various vision tasks with point supervision [3, 17, 4].

It is possible to make a quantitative comparison against methods in Group A. They solve the image classification task with extra mask annotation costs[1] to improve model robustness. Our innovation is that we achieve this effect without additional supervision costs. RRR [18] and Gradmask [20] were only tested on small-scale datasets but are replicated in the RobustViT paper [5] for ImageNet evaluation. We present a quantitative comparison in Table B with DeiT-B[2].

Our LUAB framework improves the performance on all ImageNet benchmarks, whereas Group A methods show mixed results. Importantly, unlike Group A methods, our improvements do not assume the availability of GT masks. Moreover, LUAB is applicable to general model types, while RobustViT is limited to ViT variants.

Evaluation of Group B methods is not compatible, as their target task is not image classification. We report their annotation costs for point supervision in the table. Our contribution to Group B community is the finding that *weak point supervision may be obtained without additional cost from the class labelling procedure*. OpenImagesV7 [4] introduces an efficient labelling scheme, but it relies on a pre-trained segmentation model (IRN [2]) to propose points; it is not directly comparable in our setting.

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 [1].

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.

[1] Mask: 80 & 280 sec/im, Cls: 1.13 & 36.3 sec/im for IN & COCO.

[2] ViT-B trained using the DeiT training setup [22].

				Cost (sec/im)	
Category	Approach	Target task (evaluation)	Annotation task → Annotation	ImageNet	COCO
Baseline	Classification	Image classification	cls labelling → cls labels	1.13	36.3
Ours	Classification (LUAB-Ours)	Image classification (ImageNet, COCO)	cls labelling → { cls labels AB	1.13	36.3
Group A	RRR [18], Gradmask [20], RobustViT [5]	Image classification (ImageNet evaluation in [5])	cls labelling → cls labels segmentation → object masks	1.13 80*	36.3 280*
Group B	WTP [3]	Semantic segmentation (Pascal)	cls labelling → cls labels point labelling → points	NA** NA**	NA** NA**
	UFO2 [17]	Object detection (COCO)	cls labelling → cls labels point labelling → points	NA** NA**	80*** 84.9†
	OpenImagesV7 [4]	Instance segmentation (OpenImages)	Point verification→ { cls labels points	0.8††	2.8††

Table A: **Conceptual comparison against previous work.** *It takes 80 sec/polygon [14]. ImageNet & COCO have 1 & 3.5 polygons per image, respectively. **They report results only on Pascal & COCO, respectively. ***Estimate in [17] is only theoretical and it differs from our actual time measurement of 36.3 sec/im. †Adopting [17] to the case where 1 point/cls/im is annotated. ††[4] reports 0.8 sec/click for verifying points.

Model	GT Mask	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↑
Our DeiT-B	✗	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 (Ours)	✗	+0.9	+1.6	+0.7	+5.0	+1.9	+0.5	+2.5	+2.7	+1.5	+3.3	+0.5	+1.7	+0.3	-0.8	+5.8
DeiT-B in [5]	✗	80.8	69.7	-	12.9	-	-	31.2	30.9	-	31.4	54.6	34.5	39.3	-	-
+Gradmask [20]	✓	+0.3	+0.0	-	+2.2	-	-	+0.0	+0.1	-	+2.1	+0.6	-0.4	-0.2	-	-
+RRR [18]	✓	+0.2	+0.2	-	+1.9	-	-	-0.3	+0.2	-	+2.2	+0.7	-0.1	+1.1	-	-
+RobustViT [5]	✓	-0.3	-0.6	-	+4.3	-	-	-0.3	+1.5	-	+4.5	+3.4	+2.1	+3.6	-	-

Table B: **Quantitative comparison against prior work.** We compare ours with the prior arts, including Gradmask [20], RRR [18], and RobustViT [5] using DeiT-B on ImageNet1K and variant robustness benchmarks.

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 [1].

We reject HITs based on the following criteria

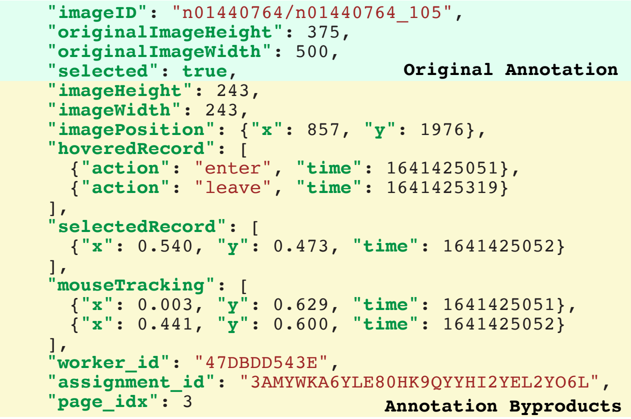
- the recall rate, defined as the proportion of retrieved classes among the existing classes, is lower on average than 0.333; or
- the accuracy of icon location, defined as the ratio of icons placed on the ground-truth class segmentation mask, is lower than 0.75; or
- the annotator has not completed 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 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 × 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. Annota-



```
    "imageID": "n01440764/n01440764_105",
    "originalImageHeight": 375,
    "originalImageWidth": 500,
    "selected": true,
    "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
```

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

tors 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 [16] 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.

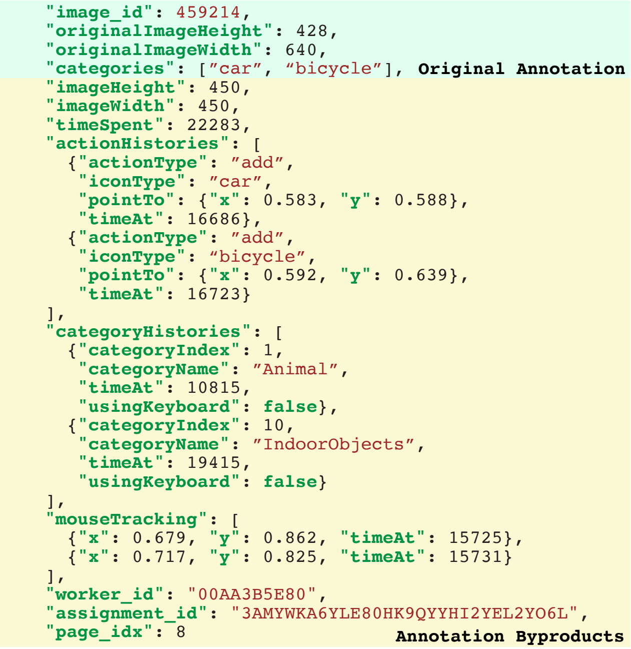
D.2. COCO-AB

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 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



```
    "image_id": 459214,
    "originalImageHeight": 428,
    "originalImageWidth": 640,
    "categories": ["car", “bicycle”],
    "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
```

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

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 ±0.118%p. The average localisation accuracy for icon placement is 92.3%, where the standard deviation is ±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.

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.

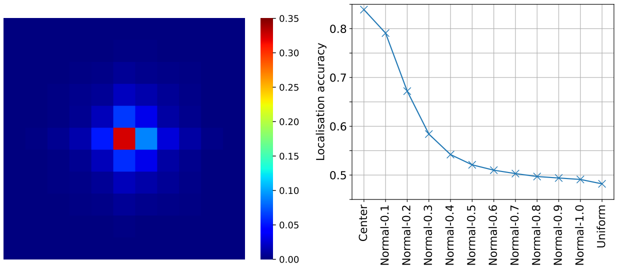


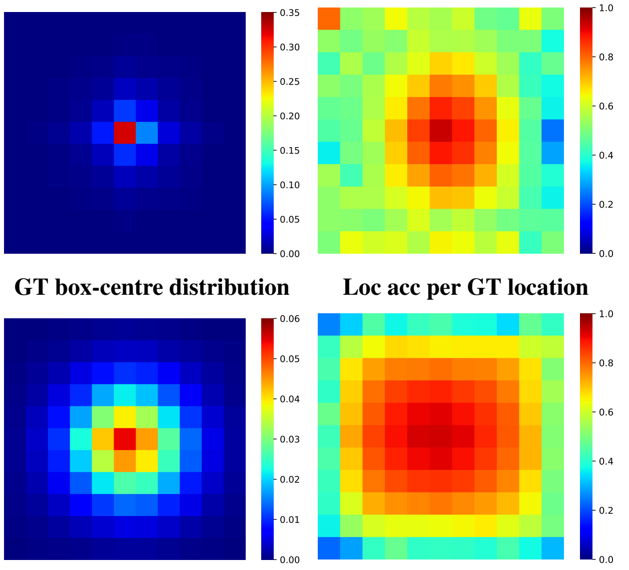
Figure C: **ImageNet GT-box statistics. Left**: distribution of GT box centres on ImageNet1K training set images. **Right**: localisation accuracy of random clicks $N((\frac{H}{2},\frac{W}{2}),\sigma^2)$. We interpolate between centre-always click ($\sigma=0$) and uniform random click ($\sigma=\infty$).

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 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

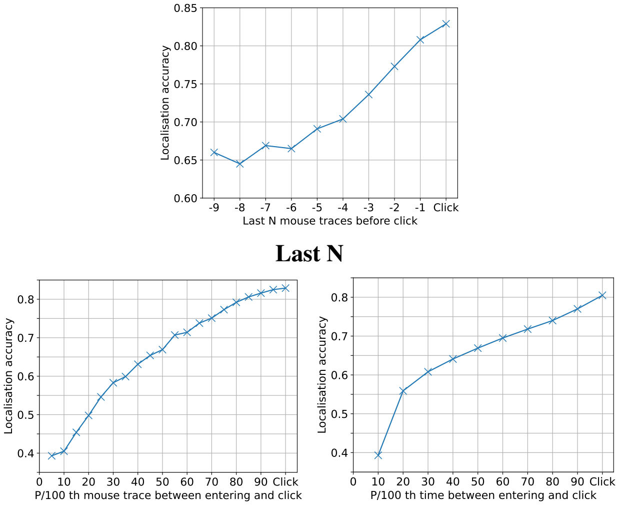


GT box-centre distribution **Loc acc per GT location**

Click distribution **Loc acc per click location**

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 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.



Last N

Trace quantile **Time quantile**

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.

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 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 [3].

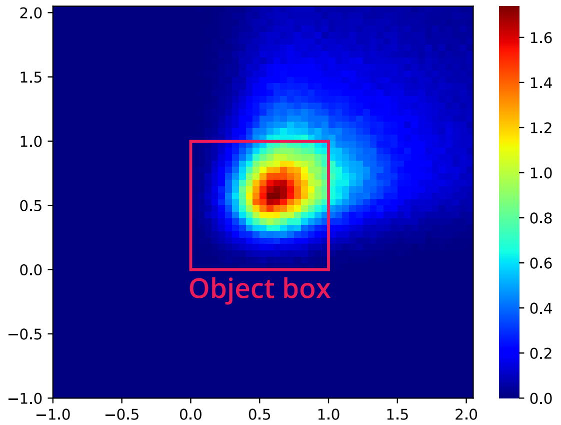


Figure F: **Click histogram relative to GT box on ImageNet.** Distribution of click positions normalised against the GT object box frame at $[0, 1] \times [0, 1]$.

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 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

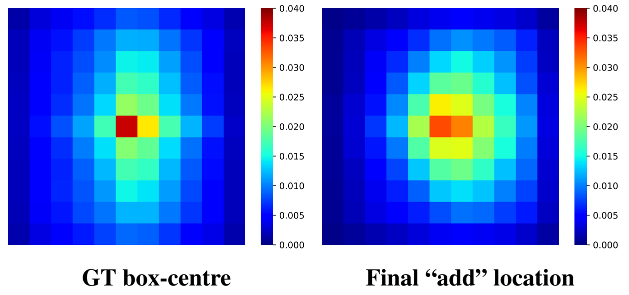


Figure G: **Statistics of icon placement.** Statistics for the location of objects and the final icon placements.

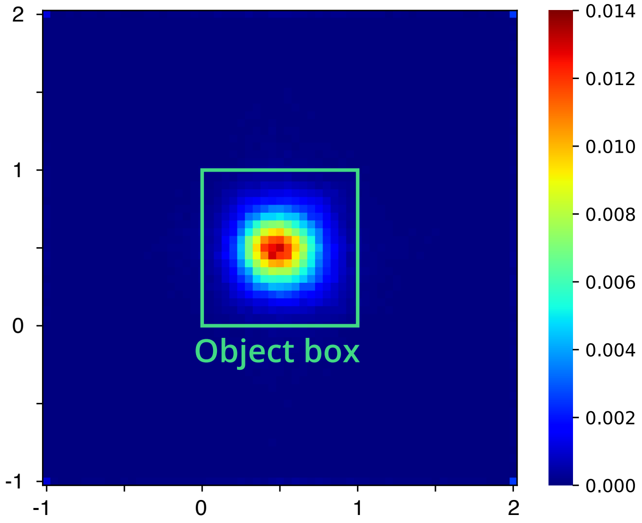


Figure H: **Icon histogram relative to the GT box on COCO.** Distribution of final “add” positions normalised against the GT object box frame at $[0, 1] \times [0, 1]$.

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.

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.

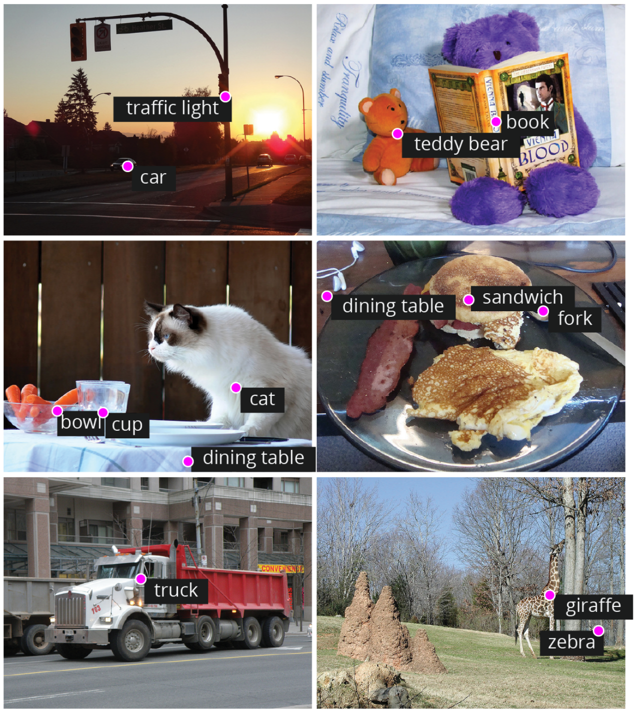


Figure I: **COCO final icon locations**. We visualise random training images; **points** are the final location of the add action for each category in actionHistories.

```
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.

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 [22] codebase[3] 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 [6] 9/0.5 (*i.e.* rand-m9-mstd0.5-inc1), Label smoothing [21] 0.1, Stochastic Depth 0.1 with the linear decay of death rate [13], and Random Erasing [11, 7] 0.25; Mixup [24] and Cutmix [23] with the probabilities 0.8 and 1.0, respectively with switching probability 0.5, and the repeated augmentation [12] 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 [9, 8, 22]. 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 $2\mathrm{e}{-5}$, 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 [11, 7] with 0.5. Specifically, we only use We use the AdamP [10] 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` generally performs 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 F 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 Table C 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.

Exploration of loss functions. Smooth ℓ^1 (Huber) loss is a natural initial baseline; it has been effective for a similar task of bounding box regression in object detection. We trained ResNet50 with the MSE and Smooth ℓ^1 loss with $\beta \in \{0.1, 1, 2\}$. The results in Table E show that MSE can be an alternative, but the Huber loss is still the best choice.

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

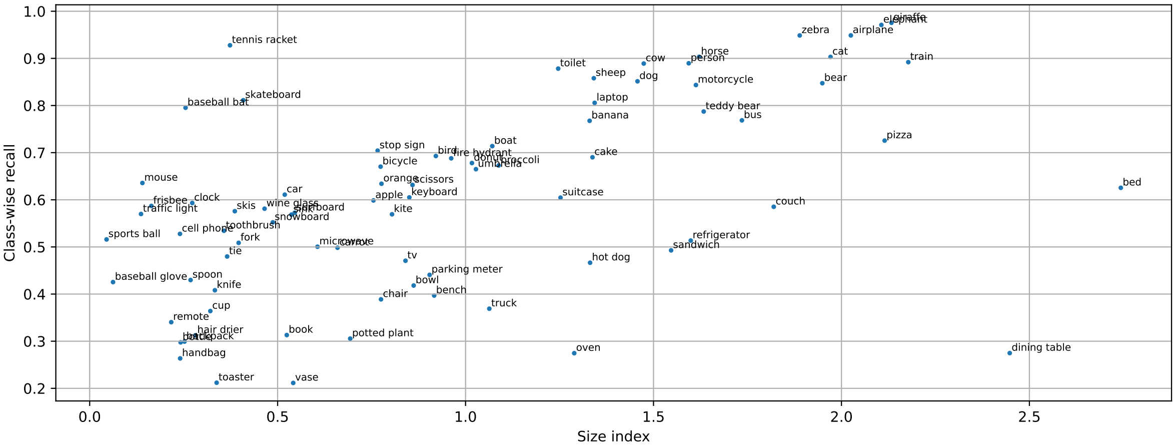


Figure K: **Recall versus size for each COCO category.**

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	79.4	**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**	**79.5**	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	83.5	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**	**83.5**	**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 C: **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	78.6	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**	**79.8**	**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	80.4	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**	**81.6**	**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	81.2	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**	**82.1**	**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 D: **Performance of ImageNet-AB on ImageNet1K without sophisticated training recipes.** We extend the study in Table **??** by training ViTs [8, 22] with simpler training recipes. We note more significant improvements due to ImageNet-AB than shown in Table **??**.

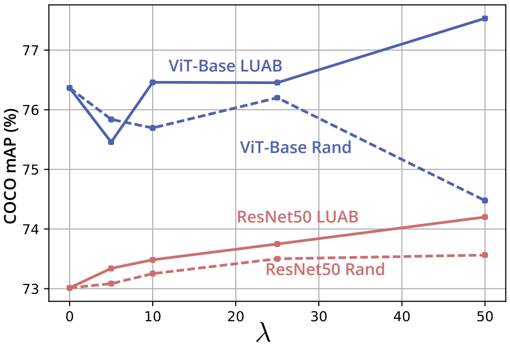


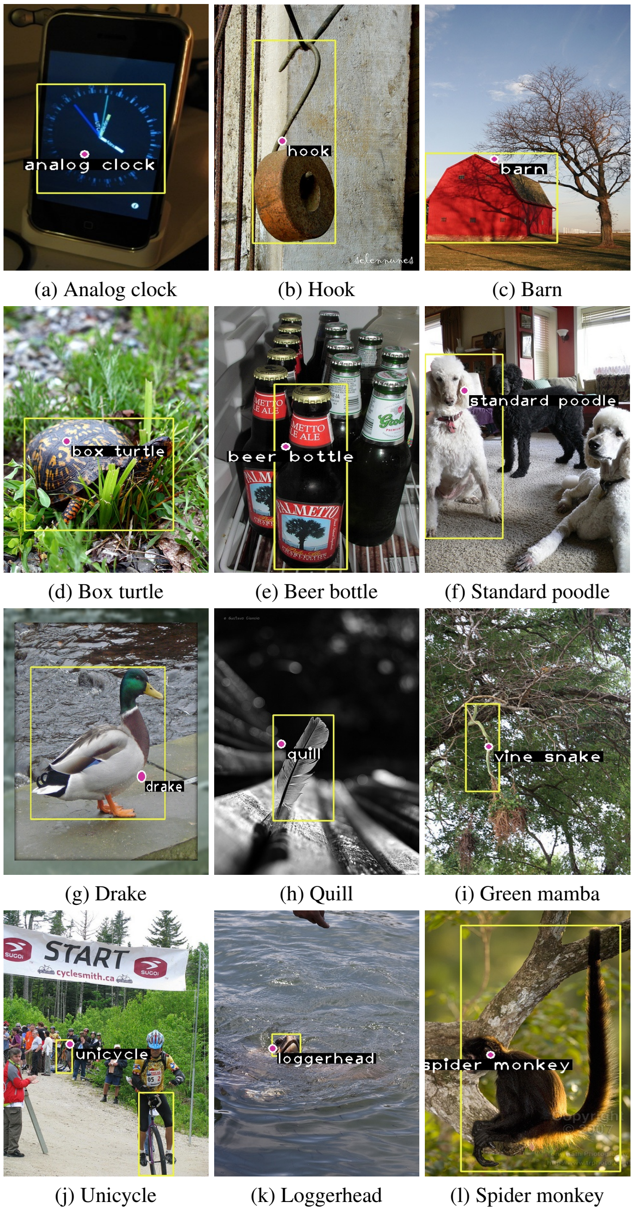
Figure L: **COCO mAP vs.** λ.

Model	IN-1K↑	IN-V2↑	IN-Real↑	ObjNet↑	SI-size↑	SI-loc↑	SI-rot↑
ℓ^1 (β=1)	77.5	65.2	**78.5**	28.5	**55.6**	**33.5**	**40.9**
ℓ^1 (β=2)	77.4	65.2	78.2	28.0	55.2	32.0	40.5
ℓ^1 (β=0.1)	76.5	64.0	77.7	27.1	53.2	30.0	38.6
MSE	**77.6**	**65.4**	78.4	**28.9**	55.5	32.6	40.7

Table E: **Exploration of loss functions for regression.**

Training data	ImageNet	+Annotation byproducts			
% Data used	100%	100%	95%	90%	80%
ImageNet1K acc (%)	72.8	**72.9**	**72.9**	72.4	71.7

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



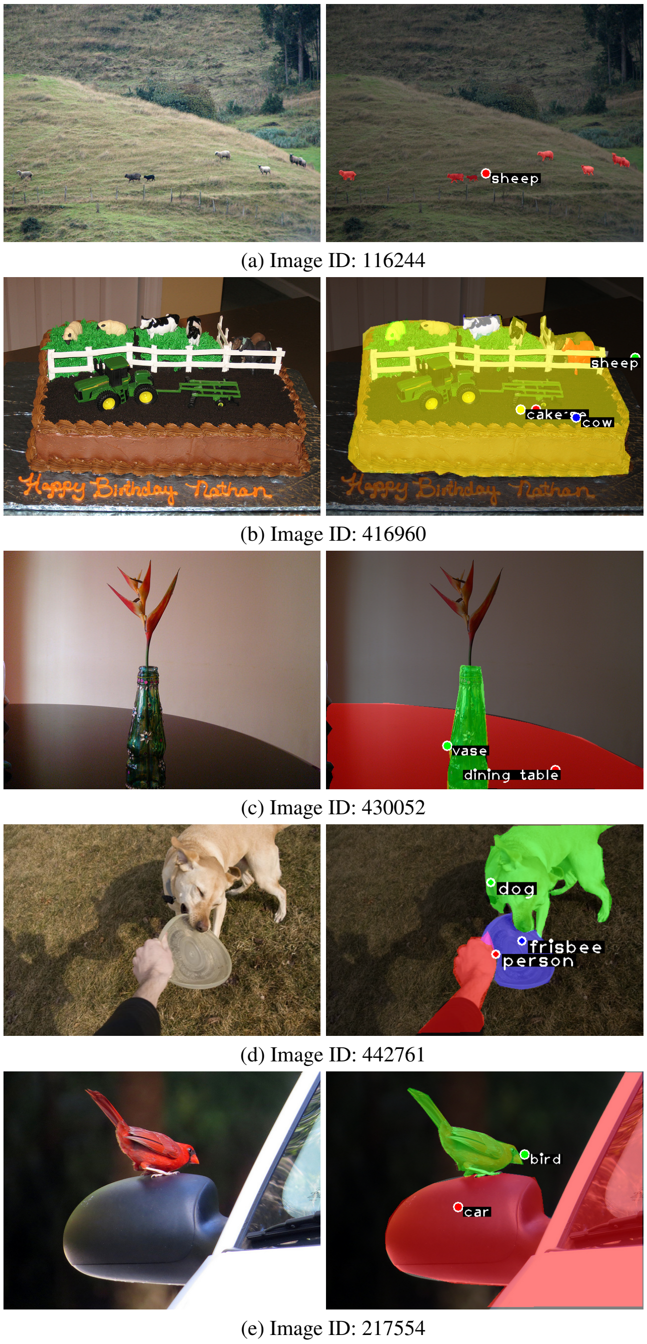
(a) Analog clock (b) Hook (c) Barn

(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 [8, 22] and require stronger augmentations. We follow the training setup provided in original ViT [8]; we limit the strong data augmentation or regularisations previously used. Table D shows the performances without strong augmentations such as RandAug [6], Stochastic Depth [13], Random Erasing [11, 7], Mixup [24], Cutmix [23] in the DeiT training regime [22]. We use a training setup similar to the one in the ViT paper [8]: 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.

References

[1] Us federal minimum wage. https://www.dol.gov/general/topic/wages/minimumwage, 2022. 1, 2

[2] Jiwoon Ahn, Sunghyun Cho, and Suha Kwak. Weakly supervised learning of instance segmentation with inter-pixel relations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2209–2218, 2019. 1

[3] 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. 1, 2, 5

[4] Rodrigo Benenson and Vittorio Ferrari. From couloring-in to pointillism: revisiting semantic segmentation supervision. In *ArXiv*, 2022. 1, 2

[5] Hila Chefer, Idan Schwartz, and Lior Wolf. Optimizing relevance maps of vision transformers improves robustness. *arXiv preprint arXiv:2206.01161*, 2022. 1, 2

[6] 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. 7, 9

[7] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 7, 9

[8] 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. 7, 8, 9

[9] 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. 7

[10] 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. 7

[11] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. 7, 9

[12] 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. 7

[13] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Weinberger. Deep networks with stochastic depth. In *European Conference on Computer Vision*, 2016. 7, 9

[14] 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. 1, 2

[15] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? *arXiv preprint arXiv:1902.10811*, 2019. 1

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

[17] 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. 1, 2

[18] 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. 1, 2

[19] 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. 1

[20] Becks Simpson, Francis Dutil, Yoshua Bengio, and Joseph Paul Cohen. Gradmask: Reduce overfitting by regularizing saliency. *arXiv preprint arXiv:1904.07478*, 2019. 1, 2

[21] 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. 7

[22] 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. 1, 7, 8, 9

[23] 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. 7, 9

[24] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 7, 9