UFO^2: A Unified Framework towards Omni-supervised Object Detection

Zhongzheng Ren[1,2*], Zhiding Yu[2], Xiaodong Yang[2*], Ming-Yu Liu[2], Alexander G. Schwing[1], and Jan Kautz[2]

[1] University of Illinois at Urbana-Champaign [2] NVIDIA

Abstract. Existing work on object detection often relies on a single form of annotation: the model is trained using either accurate yet costly bounding boxes or cheaper but less expressive image-level tags. However, real-world annotations are often diverse in form, which challenges these existing works. In this paper, we present UFO^2, a unified object detection framework that can handle different forms of supervision simultaneously. Specifically, UFO^2 incorporates strong supervision (*e.g.*, boxes), various forms of partial supervision (*e.g.*, class tags, points, and scribbles), and unlabeled data. Through rigorous evaluations, we demonstrate that each form of label can be utilized to either train a model from scratch or to further improve a pre-trained model. We also use UFO^2 to investigate budget-aware omni-supervised learning, *i.e.*, various annotation policies are studied under a fixed annotation budget: we show that competitive performance needs no strong labels for all data. Finally, we demonstrate the generalization of UFO^2, detecting more than 1,000 different objects without bounding box annotations.

Keywords: Omni-supervised, Weakly-supervised, Object Detection.

1 Introduction

State-of-the-art object detection methods benefit greatly from supervised data, which comes in the form of bounding boxes on many datasets. However, annotating images with bounding boxes is time-consuming and hence expensive. To ease this dependence on expensive annotations, '*omni-supervised learning*' [40] has been proposed: models should be trained via all types of available labeled data plus internet-scale sources of unlabeled data.

Omni-supervised learning is particularly beneficial in practice. Compared to the enormous amounts of visual data uploaded to the internet (*e.g.*, over 100 million photos uploaded to Instagram every day [1]; 300 hours of new video on YouTube each minute [2]), fully-annotated training data remains a negligible fraction. Most data is either unlabeled, or comes with a diverse set of weak labels. Hence, directly leveraging web data often requires handling labels that are incomplete, inexact, or even incorrect (noisy).

* Work partially done at NVIDIA.

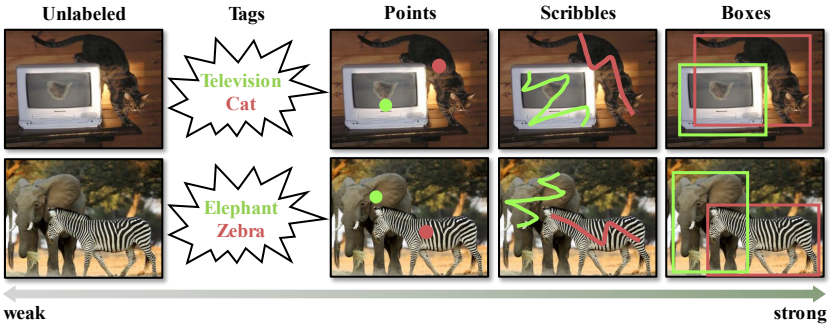

Fig. 1. Illustrative example of the supervision hierarchy.

Towards the goal of handling real-world messy data, we aim to study omni-supervised *object detection* where a plethora of unlabeled, partially labeled (with image-level class tags, points, or scribbles), and strongly labeled (with bounding boxes) images are utilized to train detection models. Examples of the considered supervisions are shown in Fig. 1. Designing a framework for omni-supervised detection is non-trivial. A big challenge is the conflict of the different architectures that have been proposed for each annotation. To address this issue, prior work either ensembles different networks trained from different annotations [25,55] or uses iterative knowledge distillation [37,50]. However, the conflict between different modules remains as it is largely addressed in a post-processing step.

In contrast, we propose UFO^2, a unified omni-supervised object detection framework that addresses the above challenges with a principled and computationally efficient solution. To the best of our knowledge, the proposed framework is the first to simultaneously handle direct supervision, various forms of partial supervision, and unlabeled data for object detection. UFO^2 (1) integrates a **unified task head** which handles various forms of supervision (Sec. 3.1), and (2) incorporates a **proposal refinement** module that utilizes the localization information contained in the labels to restrict the assignment of object candidates to class labels (Sec. 3.2). Importantly, the model is **end-to-end trainable**.

We note that assessing the efficacy of the proposed approach is non-trivial. Partial labels are hardly available in popular object detection data [30]. We thus create a simulated set of partial annotations, whose labels are synthesized to closely mimic human annotator behavior (Sec. 4). We then conduct rigorous evaluations to show that: (1) each type of label can be effectively utilized to either train a model from scratch or to boost the performance of an existing model (Sec. 5.1); (2) a model trained on a small portion of strongly labeled data combined with other weaker supervision can perform comparably to a fully-supervised model under a fixed annotation budget, suggesting a better annotation strategy in practice (Sec. 5.4); (3) the proposed model can be seamlessly generalized to utilize large-scale classification (only tags are used) data. This permits to scale the detection model to more than 1,000 categories (Appendix A).

B	T	P	S	B+U	B+T	B+T+P+S+U
[13,17,43,41,31,27]	[6,53,62,45]	[34,35]	None	[40,47,9]	[22,42,14,59,55]	**UFO^2(ours)**

Table 1. Summary of related works for object detection using different labels. (B: boxes, T: tags, P: points, S: scribbles, U: unlabeled.)

2 Related Work

In the following we first discuss related works for each single supervision type. Afterwards we introduce prior works to jointly leverage multiple labels for visual tasks. Training data usage of prior object detection works are given in Tab. 1.

Object Detection. Object detection has been one of the most fundamental problems in computer vision research. Early works [13] focus on designing hand-crafted features and multi-stage pipelines to solve the problem. Recently, Deep Neural Nets (DNNs) have greatly improved the performance and simplified the frameworks. Girshick *et al*. [16,17] leverage DNNs to classify and refine pre-computed object proposals. However, those methods are slow during inference because the proposals need to be computed online using time-consuming classical methods [56,65]. To alleviate this issue, researchers have designed DNNs that learn to generate proposals [43,20] or one-shot object detectors [31,41]. Recently, top-down solutions have emerged, re-formulating detection as key-point estimation [27]. These methods achieve impressive results. However, to train these methods, supervision in the form of accurate localization information (bounding boxes) for each object is required. Collecting this supervision is not only costly in terms of time and money, but also prevents detectors from generalizing to new environments with scarce labeled data.

Weakly-supervised Learning. Weak labels in the form of image-level category tags are studied in various tasks [26,63,49,36]. For object detection, existing works [6,53,60,45] formulate a multiple instance learning task: the input image acts as a bag of pre-computed proposals [56,3,65] and several most representative proposals are picked as detections. Bilen and Vedaldi [6] are among the first to implement the above idea in an end-to-end trainable DNN. Follow-up works boost the performance by including extra information, such as spatial relations [39,62,45] or context information [24,45]. In addition, better optimization strategies like curriculum learning [61], self-taught learning [23], and iterative refinement [39,48,15] have shown success. However, due to the limited representation ability of weak labels, these methods often suffer from two issues: (1) they cannot differentiate multiple instances of the same class when instances are spatially close; (2) they tend to focus on the most discriminative parts of an object instead of its full extent. This suggests that training object detectors solely from weak labels is not satisfactory and motivates to study a hybrid approach.

Partially-supervised Learning. Points and scribbles are two user-friendly ways of interacting with machines. Thus they are widely used in various visual

tasks such as semantic segmentation [58,29,4], instance segmentation [64], and image synthesis [38]. From a data annotation perspective, these partial labels are easier to acquire than labeling bounding boxes or masks [29]. However, partial labels are in general understudied in object detection. A few examples on this topic include Papadopoulos *et al*. [34,35] which collect click annotation for the VOC [12] dataset and train an object detector through iterative multiple instance learning. Different from their approach, however, we propose an end-to-end trainable framework and evaluate on more challenging data [30,18].

Semi-supervised Learning. Semi-supervised learning [8,33,46] aims to augment the limited annotated training set with large-scale unlabeled data to boost the performance. Recent approaches [5,32,54,57,67,66,46] on classification often utilize unlabeled data through self-training combined with various regularization techniques including consistency regularization through data augmentation [5,54,57], entropy minimization [32,28], and weight decay [5]. In this paper, we adopt the entropy regularization [32] and pseudo-labeling [28] methods to efficiently utilize unlabeled data.

For object detection, Rosenberg *et al*. [47] demonstrate that self-training is still effective. Ensemble methods [40,5] and representation learning [9,19,11,44] are shown to be useful. Nevertheless, these methods are heavily pipelined and usually assume existence of a portion of strong labels to initialize the teacher model. In contrast, our UFO^2 learns from an arbitrary combination of either strong or partial labels and unlabeled data, it is unified and end-to-end trainable.

Omni-supervised Learning. Omni-supervised learning is a more general case of semi-supervised learning in the sense that several types of available labels are mixed to train visual models jointly. Xu *et al*. [58] develop a non-deep learning method to jointly utilize image tags, partial supervision, and unlabeled data for semantic segmentation and perform competitively. Chéron *et al*. [10] extend this idea to video data by training an action localization network using various labels. However, their method cannot deal with unlabeled data.

For object detection, prior works [42,22,14,59,55] have studied to combine bounding boxes and image tags. However, these methods are either pipelined and iterative [14,22,55] or require extra activity labels and human bounding boxes to guide the detection [59]. Compared to those works, the proposed framework can handle more types of labels and, importantly, our proposed approach is end-to-end trainable.

3 UFO^2

We aim to solve omni-supervised object detection: a single object detector is learned jointly from various forms of labeled and unlabeled data. Formally, the training dataset contains two parts: an unlabeled set $\mathcal{U} = (u_i; i \in \{1, \ldots, |\mathcal{U}|\})$ and a labeled set $\mathcal{X} = (x_i; i \in \{1, \ldots, |\mathcal{X}|\})$. Each x_i is associated with one annotated label coming in one of the following four forms: (1) accurate bounding

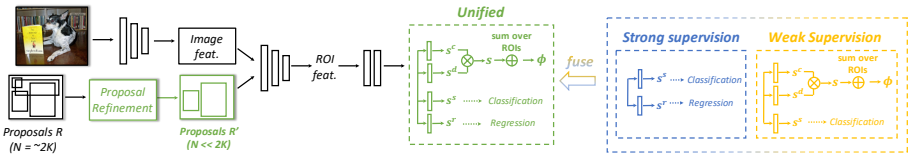

Fig. 2. The UFO^2 framework: green modules are newly proposed in this paper.

boxes, (2) a single point on the object, (3) a scribble overlaying the object in some form, or (4) image-level class tags. Note, for the first three forms of annotations we also know the semantic class. In this paper, we make **no assumptions on labels**: every form of label can make up any fraction of the training data. This is in contrast to most prior work on mixed supervision [14,55,42,59] which assumes a certain amount of strongly labeled data (bounding boxes) is always available.

Since each form of annotation has been separately studied in prior work, different frameworks have been specifically tailored for each annotation. In contrast, we present a novel unified framework UFO^2 which inherits merits of prior single supervision methods and permits to exploit arbitrary combinations of labeled and unlabeled data as shown in Fig. 2. We introduce the specific solution to handle each supervision in Sec. 3.1. We further devise an improved proposal refinement module [16,43] so as to incorporate localization information in partial labels (see Sec. 3.2).

3.1 Unified Model

As shown in Fig. 2, for a labeled input image $x \in \mathcal{X}$ or an unlabeled $u \in \mathcal{U}$, convolutional layers from an ImageNet pre-trained neural network are used to extract image features. A set of pre-computed object proposals R is refined to the set R' and then used to generate ROI features through ROI-Pooling [20]. Note that not all the proposals are used since they are usually redundant. We discuss our refinement technique in Sec. 3.2. In our proposed model, the ROI features are processed via several intermediate layers followed by a new task head as shown in Fig. 2 (center, green), which differs from classical methods.

Classical Methods. In strongly supervised frameworks [16,43], the task head consists of two fully-connected layers to produce the classification logits $s^s(r,c) \in \mathbb{R}$ for every region $r \in R'$ and class $c \in C$, and the region coordinates $s^r(r) \in \mathbb{R}^4$ for every region $r \in R'$ for bounding box regression. This is highlighted via a blue box in Fig. 2.

In weakly-supervised frameworks [6,53,45] which handle image-level tags, the task head contains three fully-connected layers to produce a class confidence score $s^c(r,c) \in \mathbb{R}$, an objectness score $s^d(r,c) \in \mathbb{R}$, and similarly, classification logits $s^s(r,c) \in \mathbb{R}$ for every region $r \in R'$ and class $c \in C$ (Fig. 2 yellow box). The class confidence score $s^c(r,c)$ and objectness score $s^d(r,c)$ are first normalized

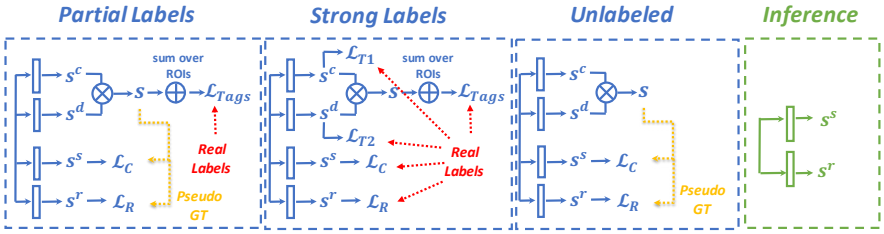

Fig. 3. Task head behavior for training (w/ partial, strong, or no labels), and inference.

via:

$$s^c(r,c) = \frac{\exp s^c(r,c)}{\sum_{c\in C} \exp s^c(r,c)}, \text{ and } s^d(r,c) = \frac{\exp s^d(r,c)}{\sum_{r\in R} \exp s^d(r,c)}. \tag{1}$$

They are then used for image-level classification. Also, $s^s(r,c)$ is used similarly for region classification using online-computed pseudo-labels.

UFO^2 Loss. We propose to fuse both heads into a unified task head to produce the four aforementioned scores simultaneously as shown in Fig. 2 (center green box). A joint objective is optimized via

$$\mathcal{L}_{\text{joint}} = \mathcal{L}_I + \frac{1}{|R'|} \sum_{r\in R', c\in C} \mathcal{L}_R(s^r(r), t(r)) + \mathcal{L}_C(s^s(r,c), y(r,c)), \tag{2}$$

where $\mathcal{L}_I$ subsumes different losses for different labels and $\mathcal{L}_C, \mathcal{L}_R$ are standard cross-entropy loss and smooth-L1 loss for region classification and regression respectively. Moreover, $y(r,c) \in \{0,1\}$ and $t(r) \in \mathbb{R}^4$ are either ground-truth region labels and regression targets from strong labels, or pseudo labels and pseudo targets generated online for partial labels and unlabeled data. We provide detailed explanations for $\mathcal{L}_I$ and how to generate pseudo labels $y(r,c)$ and pseudo targets $t(r)$ in the following. We discuss each form of annotation separately.

Tags. As illustrated in Fig. 3 left, when input images x come with image-level class tags $q(c) \in \{0,1\}$ for class $c \in C$, we neither know the exact assignment of class labels to each proposal nor the exact target location. Therefore, we first compute the image scores via $s(r,c) = s^c(r,c) \cdot s^d(r,c)$, *i.e.*, as a product of the class confidence score s^c and the objectness score s^d. Then image level evidence ϕ is obtained by summing up $s(r,c)$ across all regions: $\phi(c) = \sum_{r\in R'} s(r,c)$. We then compute $\mathcal{L}_{\text{Tags}}$ as an image-level binary cross-entropy loss for multi-label classification:

$$\mathcal{L}_{\text{Tags}}(\phi, q) = -\sum_{c\in C} q(c) \log \phi(c).$$

For samples with image-level tags we set $\mathcal{L}_I = \mathcal{L}_{\text{Tags}}$ in Eq. (2) during training. This yields semantically meaningful ROI scores $s(c,r)$, which can then be

used to generate pseudo ROI-level ground-truth to augment the training via the two region-level losses $\mathcal{L}_C$ and $\mathcal{L}_R$ as detailed in Eq. (2). We follow Ren *et al*. [45] to generate pseudo ground-truth, taking one or few diverse confident predictions.

Points & Scribbles. Similar to image-level tags, points and scribbles also don't contain the exact region-level ground-truth. However, they provide some level of localization information (*e.g*., scribbles can be very rough or accurate depending on the annotator). Therefore, we employ the same loss developed for 'Tags,' (illustrated in Fig. 3 left) but introduce extra constraints to restrict the assignment of ROIs to class labels based on the labels. Specifically, pseudo label $y(r, c) = 1$ if and only if region r contains the given point or scribble, and class c is the same as the category label of this point or scribble. These constraints filter out a lot of false-positives during training and help the framework select high quality candidate regions.

Boxes. When the input image is annotated with bounding boxes, the most naïve solution is to directly train the network using $\mathcal{L}_C$ and $\mathcal{L}_R$ losses: the real label and target are given and the scores s^s and s^r will be used for inference. Most supervised work [16,43] follows the above procedure and impressive results are achieved. Importantly, only applying these two losses in our framework will not optimize the scores s^c and s^d when learning from strong labels. However, these two scores are used as a 'teacher' to compute pseudo ground-truth for optimizing s^s and s^r when partial labels are given, as described in the previous two sections. Hence, when training with mixed annotations, we found the 'student' to be stronger than the 'teacher,' rendering weakly labeled data useless.

To address this concern, *i.e*., to enable training with mixed annotations, we found a *balanced teacher-student model* to be crucial. Specifically, for any fully labeled sample we introduce three extra losses on the latent modules, *i.e*., on s^c, s^d, ϕ, as shown in Fig. 3 second column:

$$\mathcal{L}_I = \mathcal{L}_{\text{Tags}}(\phi, q) + \frac{1}{|R|}\sum_{r\in R}(\mathcal{L}_{T1}(s^c, y, r) + \mathcal{L}_{T2}(s^d, \psi, r)). \tag{3}$$

These three losses provide a signal to the 'teacher' when using strong labels. Specifically, since s^c is normalized across all classes via a softmax, as mentioned in Eq. (1), we can naturally apply as the first strong-teacher loss a standard cross-entropy on s^c for region classification:

$$\mathcal{L}_{T1}(s^c, y, r) = -\sum_{c\in C} y'(c, r) \log s^c(c, r).$$

Hereby $y'(c, r) = 1$ for all regions r which overlap with any ground-truth boxes in class c by more than a threshold. In practice, we set this threshold to 0.5 and we use the class of the biggest overlapping ground-truth as the label if assignment conflicts occur. The second strong-teacher loss encourages the latent distribution

s^d to approach the real objectness distribution. Hence we use a KL-divergence applied on s^d:

$$\mathcal{L}_{T2}(s^d, \psi, r) = \sum_{c \in C} \psi(c, r) \log \frac{\psi(c, r)}{s^d(c, r)}.$$

Here, $\psi(c, r)$ is constructed to represent the objectness of each ROI. ψ is zero initialized and $\psi(c, r) = \text{IoU}(r, r')$ for ground-truth region r' with class c. We then normalize ψ across all $r \in R'$, following $s^d(c, r)$ normalization in Eq. (1).

In addition, we also construct an image-level class label q from the ground-truth annotations and compute the image-level classification loss $\mathcal{L}_{\text{Tags}}$ following the 'Tags' setting. This loss term improves network consistency when switching between partial labels and strong labels.

Unlabeled. For unlabeled data, we employ a simple yet effective strategy as shown in Fig. 3 third column. We use a single threshold τ on $\phi(c)$ to first pick out a set of confident classes $\hat{q}(c)$. This set of classes is used as tags to train the framework as descried in the 'Tags' section. In addition, we apply entropy regularization on s^s to encourage the model to output confident predictions on unlabeled data. The overall loss is:

$$\mathcal{L}_I = \mathcal{L}_{\text{Tags}}(\phi, \hat{q}) + H(s^s) = -\sum_{c \in C} \hat{q}(c) \log \phi(c) - \sum_{r \in R', c \in C} s^s(r, c) \log s^s(r, c),$$

where $\hat{q}(c) = \delta(\phi(c) > \tau)$ and $\delta(\cdot)$ is the delta function. As pointed out in [53,60,40], self-ensembling is helpful when utilizing unlabeled data. We thus stack multiple ROI-classification and regression layers. Pseudo ground-truth will be computed from the ROI-classification logits of one layer to supervise another one. For inference, the average prediction is adopted.

3.2 Proposal Refinement

Given strong labels, it's a standard technique [16,43] to reject most false positive proposals and re-balance the training batch using the ground-truth boxes. However, proposal refinement using partial labels has not been studied before. Specifically, we keep a specific positive and negative proposal ratio in each mini-batch. Positive proposals satisfy two requirements: (1) one of the ground-truth points or scribbles should be contained in each positive ROI; (2) all the selected positive ROIs together need to cover all the annotations. Negative proposals from the ROIs contain no labels. When generating a training batch we sample according to a pre-defined ratio. This practice dramatically decreases the number of proposals and thus simplifies subsequent optimization. We refer to the proposal set after sampling and re-balancing using R', as shown in Fig. 2 left.

4 Partial Labels Simulation

Partial labels (*e.g.*, points and scribbles) are much easier and natural to annotate than bounding boxes. They also provide much stronger localization information

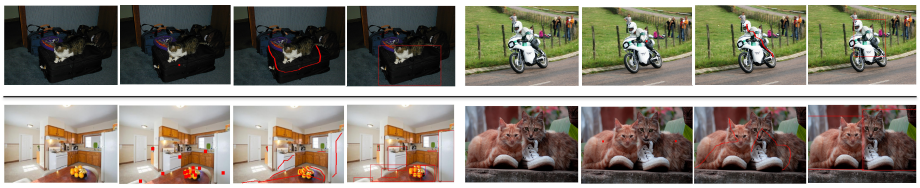

Fig. 4. Top row: labels for single instance (suitcase and person). Bottom row: labels for all the objects (see appendix for more).

compared to tags. However, these types of annotations are either incomplete (*e.g.*, part of the VOC images are labeled with points [35,34]) or missing (*e.g.*, no partial labels have been annotated for COCO or LVIS) for object detection.

As a proof-of-concept for the proposed framework, we therefore develop an approach to synthesize partial labels when ground-truth instance masks are available. It is our goal to mimic practical human labeling behavior. We are aware that the quality of the generated labels is sub-optimal. Yet these labels provide a surrogate to test and demonstrate the effectiveness of UFO^2. In this work, we generate the semantic partial labels for every object in the scene, and leave manual collection of labels to future work.

Points. When annotating points, humans tend to click close to the center of the objects [34]. However, different objects differ in shapes and poses. Hence, their center usually does not coincide with the bounding box center. To mimic human behavior, we first apply a distance transform on each instance mask. The obtained intensity maps represent the distance between the points inside the body region and the closest boundary. This distance transform usually generates a 'ridge' inside the object. We thus further normalize it and multiply with a Gaussian probability restricted to the bounding box. The final probabilistic maps are used to randomly sample one point as the annotation.

Scribbles. Scribbles are harder to simulate since human annotators generate very diverse labels. Here we provide a way to generate relatively simple scribbles. The obtained labels likely don't perfectly mimic human annotations, yet they serve as a proof-of-concept to show the effectiveness of the proposed framework. Given the instance mask, we first compute the topological skeleton, *i.e.*, a connected graph, using OpenCV's [7] skeleton function. Using this graph, we start from a random point and seek a long path by extending in both directions. At intersections we randomly choose. We post-process the paths to avoid that their ends are close to the boundary. This latter constraint is inspired by the observation that humans usually don't draw scribbles close to the boundary.

Representative generated labels are visualized in Fig. 4, where the top row shows examples for a single object (*i.e.*, suitcase and rider) and the bottom row shows the labels for all objects in the scene. We observe the partial labels to be correctly located within each object. They also exhibit great diversity in terms of location and length across different instances.

Methods	Test-scale	Label	AP	AP-50
PCL [52]	multi	tags	8.5	19.4
C-MIDN [14]	multi	tags	9.6	21.4
WSOD2 [60]	multi	tags	10.8	22.7
Ours	multi	tags	**11.4**	**24.3**
Ours	single	tags	**10.8**	**23.1**
Ours	single	points	**12.4**	**27.0**
Ours	single	scribbles	**13.7**	**29.8**
Fast-RCNN [16]	single	boxes	18.9	38.6
Faster-RCNN [43]	single	boxes	21.2	41.5
Ours	single	boxes	**25.7**	**46.3**

Table 2. Training on `COCO-80` from scratch using a single form of annotation and testing on `COCO-val`. All results are obtained with a VGG-16 backbone.

5 Experiments

We assess the proposed framework subsequently after detailing dataset, evaluation metrics and implementation.

Dataset & Evaluation Metrics. We conduct experiments on COCO [30] – the most popular dataset for object detection. Standard metrics are reported including AP (averaged over IoU thresholds) and AP-50 (IoU threshold at 50%). We use several COCO splits in this paper: (1) `COCO-80`: COCO 2014 train set of 80K images. (2) `COCO-35` (a.k.a. `valminusminival`): a 35K subset of COCO 2017 train set. (3) `COCO-115`: COCO 2017 train set, equals union of `COCO-80` and `COCO-35`. (4) `COCO-val`: COCO 2014 val set of 40K images. (5) `minival`: COCO 2017 val set of 5K images. (6) `Un-120`: COCO unlabeled set of 120k images.

Implementation Details. For a fair comparison to prior work with different forms of a single supervision, we use the most common VGG-16 and ResNet-50 backbones. SGD is used for optimization. After proposal refinement, we keep 1024 ROIs for points and scribbles and 512 for boxes as those have the most localization information and thus a reduced need for abundant ROIs.

5.1 Evaluation of Single Labels

Train From Scratch. We fist study the scenario where each single supervision is used to train a model from scratch. A VGG-16 model is trained on `COCO-80` and evaluated on `COCO-val` for a fair comparison to both weakly-supervised (tags) and strongly-supervised (boxes) work. The results are reported in Tab. 2. Following prior work, we report both single-scale and multi-scale testing results ('Test-scale' column in Tab. 2).

When using tags, our model performs comparable to prior work. We slightly increase AP and AP-50 by 0.6% and 1.6% (Tab. 2 top block). When using strong labels, our method also outperforms Fast- and Faster-RCNN baselines (Tab. 2 bottom block). We found improvements to be due to the strong teacher losses introduced in Sec. 3.1. In addition, we also report the results of partial labels

Train	Methods	backbone	Labels	AP	Extra	Labels	AP	Δ
COCO-35	ours	VGG-16	tags	4.9	COCO-80	-	5.3	**8.2%**
COCO-115	ours	VGG-16	tags	12.9	Un-120	-	13.6	**5.4%**
COCO-35	ours	ResNet-50	tags	9.8	COCO-80	-	10.5	**7.1%**
COCO-35	ours	ResNet-50	boxes	29.1	COCO-80	tags	29.4	**1.0%**
COCO-35	ours	ResNet-50	boxes	29.1	COCO-80	points	30.1	**5.5%**
COCO-35	ours	ResNet-50	boxes	29.1	COCO-80	scribbles	30.9	**6.2%**
COCO-115	ours	ResNet-50	boxes	32.7	Un-120	-	33.9	**3.7%**

Table 3. Fine-tuning to improve an existing model using each single supervision. Results are reported by testing on `minival`.

(tags, points, and scribbles) in the center block of Tab. 2, where we observe a natural correlation of performance with complexity of the labels. Note that the performance boost from tag to point (+1.6% AP/3.9%AP-50) is bigger than the boost from point to scribble (+1.3% AP/2.8%AP-50). Also, strong labels still result in the biggest performance boost: it is significantly larger than that of partial labels. Hence, bounding boxes are necessary for accurate performance.

Improve Existing Models. We now study the use of each label to boost a pre-trained object detector. Results are shown in Tab. 3. This experimental setup follows semi-supervised learning studies and mimics a common practical scenario: we want to apply a pre-trained object detector to new environments while keeping annotation cost low or while having weak labels readily available. Motivated by this scenario, pre-trained models are only fine-tuned by integrating extra weaker labels (*e.g.*, first train with boxes, and then fine-tune with points, scribbles, or unlabeled data). We study two cases in Tab. 3: (1) small scale: from COCO-35 to COCO-80 where the model sees more unlabeled data; (2) large scale: from COCO-115 to Un-120 where labeled and unlabeled data are of similar size.

In Tab. 3 (top block), we show that unlabeled data can be utilized to improve the performance of a weakly-supervised model where the VGG-16 and ResNet-50 based model are improved by 8.2% (relative) and 7.1% (relative), respectively. In Tab. 3 (center), we further demonstrate that partial labels are effective for improving a strongly-supervised model. Similarly, the relative performance improvement from tag to point (+4.5%Δ) is bigger than the improvement from point to scribble (+0.7%Δ). In Tab. 3 (bottom), we use unlabeled data for a strongly supervised ResNet-50 based model, where unlabeled data improves its performance by 1.2% AP (3.7% relative improvement).

5.2 Qualitative Results

Qualitative comparisons of the same model trained using different forms of supervision are shown in Fig. 5. From top to bottom we show predicted boxes and their confidence score when using tags, points, scribbles, and boxes. We observe stronger labels to help the model reject false positive predictions (*e.g.*, the noisy

Fig. 5. Qualitative comparison of models trained by using different labels.

small boxes on the sea and around the human head, the cars and books in the background), and also localize better true positive predictions (*e.g.*, the giraffe, surfing man, and each person in a crowd). More results and some failure modes are provided in the Appendix.

5.3 Ablation Study

Next we study the effectiveness of each proposed module in the UFO^2 framework.
Do strong teacher losses help? In Sec. 3.1, we introduce three extra losses for a balanced teacher student model when using boxes. Theses losses provide two advantages: (1) the model trained with strong labels improves as reported in Tab. 4. These results are obtained using the same setting as those given in Tab. 2. In the table, $\mathcal{L}_C + \mathcal{L}_R$ represents the vanilla version, *i.e.*, only ROI classification and regression heads are trained following supervised methods [16,43]. We then add each teacher loss and illustrate that each one of them is beneficial. The best number is achieved when all three are combined. (2) strong losses help omni-supervised learning. Without those losses, using tags, points, and scribbles will hurt the performance of a strongly pre-trained model by -5.6%, -5.2%, and -6.3% compared to the performance improvement gained in Tab. 3 (middle).
Does proposal refinement help? We evaluate the localization constraints proposed in Sec. 3.1 'Points & Scribbles' and the proposal refinement module (Sec. 3.2) following the settings of Tab. 2. The results are reported in Tab. 5. Both proposed modules improve the final performance. The localization constraints play a more important role than proposal refinement. This is reasonable as the localization constraints also consider the semantic information of partial labels.

5.4 Omni-supervised Learning

Given a fixed annotation budget, we can either choose to annotate more data with cheaper labels or less data with strong labels. With the proposed unified

Loss	$\mathcal{L}_C + \mathcal{L}_R$	$+\mathcal{L}_{T1}$	$+\mathcal{L}_{T2}$	$+\mathcal{L}_{T1} + \mathcal{L}_{T2}$	$+\mathcal{L}_{T1} + \mathcal{L}_{T2} + \mathcal{L}_{\text{Tags}}$
AP	22.6	24.8	25.1	25.5	25.7
AP-50	42.4	44.1	45.0	46.0	46.3

Table 4. Ablation study for the three strong teacher losses.

Methods	Tags	Ref_P	Ref_S	Ref_P + Con_P	Ref_S + Con_S
AP	10.8	11.1	11.6	12.4	13.7
AP-50	23.1	24.2	25.1	27.0	29.8

Table 5. Ablation study for localization constraints (Con_P and Con_S for points and scribbles) and proposal refinement (Ref_P, Ref_S).

model, we empirically study and compare several annotation policies, and we provide a new insight regarding a suitable strategy.

Annotation Time Estimation. We use the labeling time as the annotation cost and ignore other factors in this work. We approximate the annotation time of each supervision relying on the annotation and dataset statistics reported in the literature [30,4].

- **Tags:** Collecting image-level class labels takes 1 second per category according to [4]. Thus, the expected annotation time on COCO is 80 sec/img.
- **Points:** COCO [30] contains 3.5 categories and 7.7 instances per image on average. Similarly to above, it takes 1 second to annotate every non-exist classes, for $80 - 3.5 = 76.5$ seconds in total. [4] reports that annotators take a median of 2.4 seconds to click on the first instance of a class, and 0.9 seconds for every additional instances. Thus the total labeling time is $76.5 + 3.5 \times 2.4 + (7.7 - 3.5) \times 0.9 = 88.7$ sec/img. Note that point supervision is only 1.1 times more expensive than tags which is very efficient.
- **Scribbles:** For each existing class, drawing a free-form scribble takes 10.9 seconds on average [29,4]. Hence, the total time is $76.5 + 7.7 \times 10.9 = 160.4$ sec/img. This number is roughly twice the time of labeling tags or points.
- **Boxes:** It took 35s for one high quality box according to [51]. Hence, the total annotation time is $76.5 + 7.7 \times 35 = 346$ sec/img.

Given above approximations, we roughly know that annotating 1 image with bounding boxes takes as much time as annotating 4.33/3.9/2.16 images with tags/points/scribbles.

Budget-aware Omni-supervised Detection. We wonder: *what annotation policy maximizes performance given a budget?* Let's assume the total budget is fixed, *e.g.*, 800,000 seconds. We empirically study several policies as listed below: (1) `MOST`: we aim to maximize the number of images thus the entire budget is used to acquire tag annotation; (2) `STRONG`: all the budget is used to annotate bounding boxes, which is widely-adopted in practice; (3) `EQUAL`: use one quarter of the budget for each label; (4) `EQUAL-NUM`: same amount labeled for each.

Via above labeling time analysis and single-label experiments, we find that annotating points is a good choice among partial labels: roughly as efficient as tags but leads to better results; only half the price of scribbles but performs

Policy	Image Amount	Labels	AP
MOST	10000	T	3.0 ± 0.57
STRONG	2312 + 7688	B+U	13.97 ± 0.98
EQUAL	2500 + 2255 + 1250 + 578 + 5417	T+P+S+B+U	5.87 ± 0.70
EQUAL-NUM	1185 × 4 + 5260	T+P+S+B+U	9.43 ± 0.68
80%B	1804 + 1850 + 6346	P+B+U	**14.11 ± 1.01**
50%B	4510 + 1156 + 4334	P+B+U	11.13 ± 1.12
20%B	7215 + 462 + 2323	P+B+U	4.47 ± 0.75

Table 6. Budget-aware Omni-supervised Detection (T: tags, P: points, S: scribbles, B: boxes, U: unlabeled). Mean and standard deviation of Average Precision (AP) are reported over three runs.

comparably well. We thus also study scenarios with different combinations of boxes and points: (1) 80%B: 80% budget on boxes; 20% on points; (2) 50%B: 50% budget on boxes; 50% on points; (3) 20%B: 20% budget on boxes; 80% on points.

As reported in Tab. 6, for the fixed budget of 800,000s, we first 'annotate' (sample from COCO-35) 10,000 images with tags for the MOST policy. The other settings will only annotate less images. Annotations are then sampled from those 10,000 images. For example, STRONG will annotate 2,312 images with boxes and the rest remains unlabeled, which will also be utilized in our method given in Sec. 3.1. Therefore, training will use the same 10,000 images, albeit different policies make use of different labels. A VGG-16 based model is trained as described above and evaluated on minival.

We observe: (1) strong labels are still very important and the policy STRONG outperforms other popular polices by a great margin (Tab. 6 top half). (2) It's not necessary to annotate every image with boxes to achieve competitive results. 80%B is slightly better than STRONG and 50%B also performs better than EQUAL-NUM. This result suggests that spending a certain amount of cost annotating more images with points is a better annotation strategy than the commonly adopted bounding box annotation (STRONG).

6 Conclusions

We present UFO^2, a novel unified framework for omni-supervised object detection. It handles strong labels, several forms of partial annotations (tags, points, and scribbles), and unlabeled data simultaneously. UFO^2 is able to utilize each label effectively, permitting to study budget-aware omni-supervised object detection. We also assess a promising annotation policy.

Acknowledgement: ZR is supported by Yunni & Maxine Pao Memorial Fellowship. This work is supported in part by NSF under Grant No. 1718221 and MRI #1725729, UIUC, Samsung, 3M, Cisco Systems Inc. (Gift Award CG 1377144) and Adobe.

References

1. Instagram statistics 2019. www.omnicoreagency.com/instagram-statistics/ 1
2. Youtube statistics 2019. https://merchdope.com/youtube-stats/ 1
3. Arbeláez, P., Pont-Tuset, J., Barron, J., Marques, F., Malik, J.: Multiscale combinatorial grouping. In: CVPR (2014) 3
4. Bearman, A.L., Russakovsky, O., Ferrari, V., Li, F.: What's the point: Semantic segmentation with point supervision. ECCV (2016) 4, 13
5. Berthelot, D., Carlini, N., Goodfellow, I.J., Papernot, N., Oliver, A., Raffel, C.: Mixmatch: A holistic approach to semi-supervised learning. In: NeurIPS (2019) 4
6. Bilen, H., Vedaldi, A.: Weakly supervised deep detection networks. In: CVPR (2016) 3, 5
7. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000) 9
8. Chapelle, O., Schölkopf, B., Zien, A. (eds.): Semi-Supervised Learning. The MIT Press (2006) 4
9. Chen, Y., Li, W., Sakaridis, C., Dai, D., Gool, L.V.: Domain adaptive faster R-CNN for object detection in the wild. In: CVPR (2018) 3, 4
10. Chéron, G., Alayrac, J.B., Laptev, I., Schmid, C.: A flexible model for training action localization with varying levels of supervision. In: NIPS (2018) 4
11. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: ICCV (2015) 4
12. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. In: IJCV (2010) 4
13. Felzenszwalb, P.F., Girshick, R.B., McAllester, D.A., Ramanan, D.: Object detection with discriminatively trained part-based models. T-PAMI (2010) 3
14. Gao, Y., Liu, B., Guo, N., Ye, X., Wan, F., You, H., Fan, D.: C-midn: Coupled multiple instance detection network with segmentation guidance for weakly supervised object detection. In: ICCV (2019) 3, 4, 5, 10
15. Ge, W., Yang, S., Yu, Y.: Multi-evidence filtering and fusion for multi-label classification, object detection and semantic segmentation based on weakly supervised learning. In: CVPR (2018) 3
16. Girshick, R.B.: Fast R-CNN. In: ICCV (2015) 3, 5, 7, 8, 10, 12
17. Girshick, R.B., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR (2014) 3
18. Gupta, A., Dollar, P., Girshick, R.: Lvis: A dataset for large vocabulary instance segmentation. In: CVPR (2019) 4, 18
19. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR (2019) 4
20. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: ICCV (2017) 3, 5
21. Hu, R., Dollár, P., He, K., Darrell, T., Girshick, R.: Learning to segment every thing. In: CVPR (2018) 18
22. Inoue, N., Furuta, R., Yamasaki, T., Aizawa, K.: Cross-domain weakly-supervised object detection through progressive domain adaptation. In: CVPR (2018) 3, 4
23. Jie, Z., Wei, Y., Jin, X., Feng, J., Liu, W.: Deep self-taught learning for weakly supervised object localization. In: CVPR (2017) 3
24. Kantorov, V., Oquab, M., Cho, M., Laptev, I.: Contextlocnet: Context-aware deep network models for weakly supervised localization. In: ECCV (2016) 3
25. Khodabandeh, M., Vahdat, A., Ranjbar, M., Macready, W.G.: A robust learning approach to domain adaptive object detection. In: ICCV (2019) 2

26. Khoreva, A., Benenson, R., Hosang, J., Hein, M., Schiele, B.: Simple does it: Weakly supervised instance and semantic segmentation. In: CVPR (2017) 3
27. Law, H., Deng, J.: Cornernet: Detecting objects as paired keypoints. In: ECCV (2018) 3
28. Lee, D.H.: Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. ICML 2013 Workshop (2013) 4
29. Lin, D., Dai, J., Jia, J., He, K., Sun, J.: Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. CVPR (2016) 4, 13
30. Lin, T., Maire, M., Belongie, S.J., Bourdev, L.D., Girshick, R.B., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. CoRR (2014) 2, 4, 10, 13
31. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: ECCV (2016) 3
32. Miyato, T., Maeda, S., Koyama, M., Ishii, S.: Virtual adversarial training: A regularization method for supervised and semi-supervised learning. T-PAMI (2019) 4
33. Oliver, A., Odena, A., Raffel, C.A., Cubuk, E.D., Goodfellow, I.: Realistic evaluation of deep semi-supervised learning algorithms. NeurIPS (2018) 4
34. Papadopoulos, D.P., Uijlings, J.R.R., Keller, F., Ferrari, V.: Extreme clicking for efficient object annotation. In: ICCV (2017) 3, 4, 9
35. Papadopoulos, D.P., Uijlings, J.R.R., Keller, F., Ferrari, V.: Training object class detectors with click supervision. In: CVPR (2017) 3, 4, 9
36. Papandreou, G., Chen, L., Murphy, K.P., Yuille, A.L.: Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In: ICCV (2015) 3
37. Pardo, A., Xu, M., Thabet, A.K., Arbelaez, P., Ghanem, B.: BAOD: budget-aware object detection. CoRR **abs/1904.05443** (2019) 2
38. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: CVPR (2019) 4
39. Peng, X., Sun, B., Ali, K., Saenko, K.: Learning deep object detectors from 3d models. In: ICCV (2015) 3
40. Radosavovic, I., Dollár, P., Girshick, R.B., Gkioxari, G., He, K.: Data distillation: Towards omni-supervised learning. In: CVPR (2018) 1, 3, 4, 8
41. Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR (2016) 3
42. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: CVPR (2017) 3, 4, 5
43. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: TPAMI (2016) 3, 5, 7, 8, 10, 12
44. Ren, Z., Lee, Y.J.: Cross-domain self-supervised multi-task feature learning using synthetic imagery. In: CVPR (2018) 4
45. Ren, Z., Yu, Z., Yang, X., Liu, M.Y., Lee, Y.J., Schwing, A.G., Kautz, J.: Instance-aware, context-focused, and memory-efficient weakly supervised object detection. In: CVPR (2020) 3, 5, 7
46. Ren*, Z., Yeh*, R.A., Schwing, A.G.: Not all unlabeled data are equal: Learning to weight data in semi-supervised learning. In: arXiv:2007.01293 (2020), * equal contribution 4
47. Rosenberg, C., Hebert, M., Schneiderman, H.: Semi-supervised self-training of object detection models. In: WACV/MOTION (2005) 3, 4
48. Shen, Y., Ji, R., Zhang, S., Zuo, W., Wang, Y.: Generative adversarial learning towards fast weakly supervised detection. In: CVPR (2018) 3

49. Singh, G., Saha, S., Sapienza, M., Torr, P., Cuzzolin, F.: Online real time multiple spatiotemporal action localisation and prediction on a single platform. In: ICCV (2017) 3
50. Singh, K.K., Xiao, F., Lee, Y.J.: Track and transfer: Watching videos to simulate strong human supervision for weakly-supervised object detection. In: CVPR (2016) 2
51. Su, H., Deng, J., Fei-Fei, L.: Crowdsourcing annotations for visual object detection. In: AAAI Technical Report, 4th Human Computation Workshop (2012) 13
52. Tang, P., Wang, X., Bai, S., Shen, W., Bai, X., Liu, W., Yuille, A.: PCL: Proposal cluster learning for weakly supervised object detection. In: T-PAMI (2018) 10
53. Tang, P., Wang, X., Bai, X., Liu, W.: Multiple instance detection network with online instance classifier refinement. In: CVPR (2017) 3, 5, 8
54. Tarvainen, A., Valpola, H.: Weight-averaged consistency targets improve semi-supervised deep learning results. NeurIPS (2017) 4
55. Uijlings, J.R.R., Popov, S., Ferrari, V.: Revisiting knowledge transfer for training object class detectors. In: CVPR (2018) 2, 3, 4, 5
56. Uijlings, J., van de Sande, K., Gevers, T., Smeulders, A.: Selective search for object recognition. IJCV (2013) 3
57. Xie, Q., Dai, Z., Hovy, E., Luong, M.T., Le, Q.V.: Unsupervised data augmentation for consistency training. arXiv:1904.12848 (2019) 4
58. Xu, J., Schwing, A.G., Urtasun, R.: Learning To Segment under Various Forms of Weak Supervision. In: CVPR (2015) 4
59. Yang, Z., Mahajan, D., Ghadiyaram, D., Nevatia, R., Ramanathan, V.: Activity driven weakly supervised object detection. In: CVPR (2019) 3, 4, 5
60. Zeng, Z., Liu, B., Fu, J., Chao, H., Zhang, L.: Wsod2: Learning bottom-up and top-down objectness distillation for weakly-supervised object detection. In: ICCV (2019) 3, 8, 10
61. Zhang, X., Feng, J., Xiong, H., Tian, Q.: Zigzag learning for weakly supervised object detection. In: CVPR (2018) 3
62. Zhang, Y., Bai, Y., Ding, M., Li, Y., Ghanem, B.: W2f: A weakly-supervised to fully-supervised framework for object detection. In: CVPR (2018) 3
63. Zhou, B., Khosla, A., A., L., Oliva, A., Torralba, A.: Learning Deep Features for Discriminative Localization. CVPR (2016) 3
64. Zhou, X., Zhuo, J., Krähenbühl, P.: Bottom-up object detection by grouping extreme and center points. In: CVPR (2019) 4
65. Zitnick, C.L., Dollár, P.: Edge boxes: Locating object proposals from edges. In: ECCV (2014) 3
66. Zou, Y., Yu, Z., Liu, X., Kumar, B., Wang, J.: Confidence regularized self-training. In: ICCV (2019) 4
67. Zou, Y., Yu, Z., Vijaya Kumar, B., Wang, J.: Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: ECCV (2018) 4