Narrowing the Gap: Improved Detector Training With Noisy Location Annotations

Shaoru Wang, Jin Gao, Bing Li, and Weiming Hu, *Senior Member, IEEE*

Abstract— Deep learning methods require massive of annotated data for optimizing parameters. For example, datasets attached with accurate bounding box annotations are essential for modern object detection tasks. However, labeling with such pixel-wise accuracy is laborious and time-consuming, and elaborate labeling procedures are indispensable for reducing man-made noise, involving annotation review and acceptance testing. In this paper, we focus on the impact of noisy location annotations on the performance of object detection approaches and aim to, on the user side, reduce the adverse effect of the noise. First, noticeable performance degradation is experimentally observed for both one-stage and two-stage detectors when noise is introduced to the bounding box annotations. For instance, our synthesized noise results in performance decrease from 38.9% AP to 33.6% AP for FCOS detector on COCO test split, and 37.8%AP to 33.7%AP for Faster R-CNN. Second, a self-correction technique based on a Bayesian filter for prediction ensemble is proposed to better exploit the noisy location annotations following a Teacher-Student learning paradigm. Experiments for both synthesized and real-world scenarios consistently demonstrate the effectiveness of our approach, *e.g.*, our method increases the degraded performance of the FCOS detector from 33.6% AP to 35.6% AP on COCO.

Index Terms— Object detection, noisy label, Bayesian estimation, teacher-student learning.

I. INTRODUCTION

OBJECT detection aims at localizing individual objects and recognizing their categories. The imaginary bounding boxes are usually adopted to represent the object locations

Shaoru Wang, Jin Gao, and Bing Li are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100190, China (e-mail: wangshaoru2018@ia.ac.cn; jin.gao@nlpr.ia.ac.cn; bli@nlpr.ia.ac.cn).

Weiming Hu is with the CAS Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Shanghai 200031, China, also with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100190, China (e-mail: wmhu@nlpr.ia.ac.cn).

for the sake of simplicity. Considerable large-scale datasets, including PASCAL VOC [1], COCO [2], and Objects365 [3], contribute to rapid advances in modern CNN-based object detection [4], [5], [6], [7], [8], [9] greatly. These datasets focus on pre-defined categories, and all object instances belonging to these categories are labeled with bounding boxes together with the corresponding categories.

These objects' boundary coordinates are typically annotated with pixel-wise accuracy, which means annotation skills training for annotators is required before starting annotating and an explicit verification step is indispensable for achieving fine-grained annotation quality [2]. What's more, zoom-in operations are sometimes required for annotators to watch the distinct small objects better, which makes such labor-intensive annotations even more time-consuming. So the inevitable negligence while annotating a huge dataset like COCO (which contains 143k images and 860k object annotations) may cause man-made noise involved in the annotations. Since detection has been widely applied in many different varieties of niche and emerging areas, there have been large amounts of public and private datasets for different AI-driven applications. However, not all teams can afford such huge labor or time cost and organize such a professional processing line for annotation as the COCO team has done. In some cases, one may possess a limited budget or face an impending deadline. The question is whether such fine-grained annotations like COCO are essential for the detection task, what will happen to the detectors when the annotations are corrupted by some noise due to the limited budget or impending deadline, and whether appropriate relaxation on the strict annotation standard is acceptable. Some existing studies have focused on learning with noisy category labels ([10], [11], [12], and [13]), and the adverse effects on both synthesized and real-world noisy datasets have been vastly investigated. In these studies, massive effective methods are also proposed to learn robust classification models against label noise. Nevertheless, there are seldom studies focused on the noise in the location annotations for the detection tasks. How the quality of location annotations affects the performance of detectors is rarely studied.

In this paper, we first investigate the impact of noisy location annotations in the detection task, and non-negligible performance degradation is observed (see Fig. 1) when the very plain Gaussian noise is injected into the location annotations of detection datasets. We conduct all of these benchmark-simulated label noise evaluations on COCO [2] for its popularity, which involves about 118k images for training
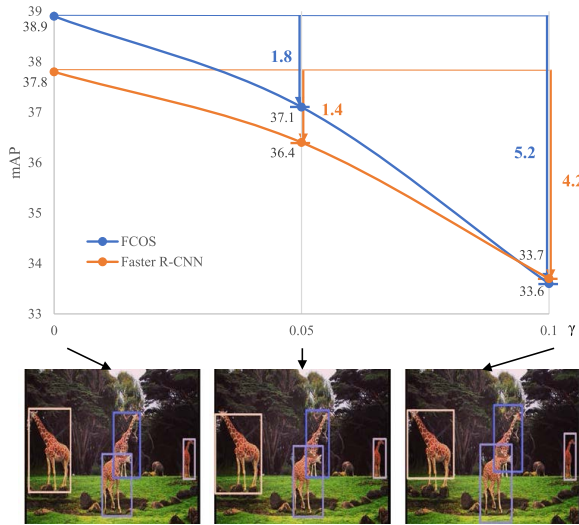
Fig. 1. Noticeable degradation of performance is observed when the location annotations of the dataset used for training are corrupted by noise. The bottom figures illustrate the location annotations under different noise levels.

(`train`), 5k for validation (`minival`) and 20k reserved for testing (`test-dev`). We simply add Gaussian noise to each of the box's boundary coordinates independently for each instance in the `train` split. We choose FCOS [9] and Faster R-CNN [5], two of the representative single-stage and two-stage detectors respectively, for evaluation. We adopt the original default training configuration of these methods, except the replacement of the annotations of the `train` split with our corrupted ones. In Fig. 1, the impact of different noise levels in terms of performance degradation is evaluated on the original COCO `test-dev` split, where γ is a parameter used to control the level of the noise and the higher one indicates more severe noisy corruption. Details of the way to generate the noise are presented in Sec. III-B. The results indicate that the detection approaches without particular treatment possibly have a poor ability to learn robust models against the noise in location annotations. We attribute it to the fact that most of these methods blindly assume the annotations to be accurate and credible. To further show the impact in the real-world scenario, we manually labeled a subset of the COCO dataset, up to 12k images, with a rather loosened label standard, contributing to about 90k rough bounding-box annotations. Performance degradation is also observed on this real-world annotation-noise subset, as shown in Sec. III-B. Based on the above analyses, we insist that fine-grained location annotations with pixel-wise accuracy are crucial to the real-world applications of these detectors.

However, noise corruption is inevitable or incurred yet in some cases. So it is vital to study and develop effective methods to alleviate the adverse effect of the noisy location annotations in the detection task. In this paper, a simple and efficient method is proposed to make effective use of noisy annotations among large datasets. It is known that the ensemble methods [14] may have the ability to integrate multiple predictions from different models and get closer to the real noise-free labels. We extend this idea to the ensemble of predictions from an individual detector. To this

end, an ensemble method based on a Bayesian filter is proposed, which can correct the noisy box annotations by the predictions of the detectors themselves. This integration considers both the probability of the prediction belonging to the corresponding object and its classification confidence. Furthermore, to avoid the trivial solution during the direct use of this self-correction process in training, a Teacher-Student learning framework is adopted. Specifically, a teacher detector is first trained on the noisy dataset and then used to correct box coordinate noise by prediction ensemble. The student detector is then trained based on the corrected annotations and we achieve better performance as expected in comparison with the frozen teacher detector. In other words, we boost the student detector's locating performance by exploiting the self-correction ability of the noisy teacher detector.

In summary, our contributions are three-fold:
- Some insightful analyses on the impact of noisy location annotations for the detection task are presented and noticeable performance degradation is observed in both synthesized and real-world scenarios.
- A novel prediction ensemble approach based on a Bayesian filter is proposed, which enables better exploitation of noisy annotations in a Teacher-Student learning framework.
- Our method provides a simple baseline for the study of noisy location annotations, which is expected to advance the research related to localization tasks, *e.g.*, object detection, tracking, *etc.*

The code and our newly-annotated noisy subset will be made publicly available at https://github.com/wangsr126/NDet to promote the study on this topic.

II. RELATED WORKS

A. Object Detection

Modern deep-learning-based methods for object detection can be generally formulated as multi-region classification and regression tasks. First, a large number of anchors are tiled on the image, and each of them is assigned to one of the ground-truth objects in the image or just the background. Then the category and relative coordinates to the specified nearby ground-truth object (if exists) are predicted for each anchor [5], [6], [7]. A class of methods cascades the above procedures several times to refine predictions, which are commonly known as *two-stage* methods [5], [15]. Another interesting direction simplifies the anchor from the box to a single point, leading to a simpler paradigm and more potential in terms of generalization ability [9], [16], namely *anchor-free* methods. In this paper, we focus on the regression branch for the above detectors. Relative coordinates from the anchor[1] to the ground-truth box of the assigned object are predicted in the regression branch. The L_p-norm-based loss (*e.g.*, smooth L_1 loss [5], L_2 loss [4], [17], L_1 loss [18]) and IoU-based loss (*e.g.*, IoU loss [19], GIoU loss [20]) are usually adopted for supervision. However, these methods assume the annotations of location are absolutely accurate, without

---

[1] Anchor box and anchor point are collectively referred to as anchor in the following part without loss of generality.

modeling the noise in the annotations, which shows poor anti-noise performance. KL-loss [21] considers the inherent ambiguities of the ground-truth bounding boxes, but it also assumes the annotations are accurate for most instances, which is thus orthogonal to our case.

B. Label Noise

Corrupted labels are ubiquitous in many real-world collected and annotated datasets, and can severely impair the performance of deep neural networks especially when the dataset is annotated with a limited budget. The label noise may arise upon mistakes of human annotators or automatic label extraction tools. For classification tasks, several approaches have sought to promote the robustness of classifiers to label corruption. One representative class of methods aims to reduce noisy samples' impact with carefully designed losses [12], [22], [23], regularization terms [24], [25] or adaptive sample re-weighting [13], [26], [27], [28]. Another direction focuses on improving the label quality by correcting the noisy labels [10], [11], [29], [30]. Most of the above studies usually conduct experiments on noisy datasets generated from CIFAR-10, CIFAR-100 by artificially corrupting the true labels. Few of them [12], [22] conduct extensive experiments on real-world noisy datasets for classification tasks.

Departing from them, our approach mainly focuses on correcting the noisy location annotations in object detection tasks, which bears some similarities to the above trend of correcting noisy labels. Some of the methods designed for classification tasks can be naturally extended to detection tasks [10], [31] to solve noisy category labels, *e.g.*, mistaken labels or missing labels. But little attention has been paid to the noise in location annotations. Some works for detection focus on semi-supervised settings, where only a subset of data are annotated while others remain unlabeled [32], [33], [34], or weakly-supervised settings, where only image-level category tags are available [35], [36], [37], [38]. All of these paradigms show significant differences to our settings. The most related works to ours are the noisy location annotation study conducted by Mao et al. [39] and some followers [40], [41]. In contrast to these works, we additionally analyze the pattern of human-made label noise and the generalization performance to the real-world noisy dataset, which shows more practical significance.

C. Teacher-Student Learning

Teacher-Student learning is a learning paradigm that uses a teacher network to train a student network for various vision tasks. The idea is introduced in knowledge distillation [42], in which a teacher model is generally pre-trained on a specific task and kept frozen during distillation. Knowledge distillation is widely adopted in various tasks, *e.g.*, classification [42], [43], [44], [45], detection [46], [47], and segmentation [48], [49]. The Teacher-Student learning framework is also adopted in semi-supervised learning [34], [50], [51]. For instance, Mean Teacher [50] constructs the teacher network as a moving average of student network and the predictions of the teacher are seen as pseudo labels for the student to learn

against. Besides, it is also widely used in self-supervised learning [52], domain adaptation [53], *etc.* Our Teacher-Student framework bears some similarities to knowledge distillation. However, the improvement of the student in distillation comes from the additional knowledge of the teacher, while the teacher itself in our method may not show superiority to the student. The improvement in our method is derived from our newly-designed annotation correction module, which can serve as a better supervision for the student detector. Hopefully, our framework is compatible with other distillation methods, though it is beyond the scope of this paper.

III. METHODOLOGY

In this section, we first briefly describe the modern detectors as preliminary. Then some experimental analyses on the impact of noisy location annotations are conducted. Finally, a simple yet effective self-correction technique is proposed based on the prediction ensemble paradigm, which can effectively improve the quality of location annotations. Along with the Teacher-Student learning framework, our method can practically narrow the performance gap between detectors with and without noisy location annotations.

A. Preliminary on Object Detection

Most modern detectors [5], [6], [7], [9] follow a dense prediction paradigm. First, a large number of anchors are tiled on the image. During the training phase, each of them is assigned to one of the objects in the image or just to the background, and fancy losses are designed to guide them to predict the right category and regress the relative coordinates to the specified nearby ground-truth object (if exists). The above process can be formulated as follows.

For each anchor a_i, classification scores $\boldsymbol{p}_i$ and parameterized coordinates $\boldsymbol{t}_i$ are predicted, where, $\boldsymbol{p}_i$ is usually a C-dimension vector, representing probabilities of belonging to the predefined C categories, and $\boldsymbol{t}_i$ is a 4-dimension vector, usually representing the relative offsets from the anchor to the predicted bounding box. Then a_i is assigned to one of the objects in the image or just to the background, consequently constructing the class label c_i^* and regression target $\boldsymbol{t}_i^*$. The latter is encoded based on the assigned ground truth box $\boldsymbol{b}_j^*$, and $\boldsymbol{b}_j^* = [l_j^*, r_j^*, t_j^*, b_j^*]$ for four object bound coordinates. The following training loss function is applied on these predictions:

$$L(\{\boldsymbol{p}_i\}, \{\boldsymbol{t}_i\}) = \lambda_{cls} \sum_i L_{cls}(\boldsymbol{p}_i, c_i^*)$$
$$+ \lambda_{reg} \sum_i \mathbb{I}_{\{c_i^*>0\}} L_{reg}(\boldsymbol{t}_i, \boldsymbol{t}_i^*), \quad (1)$$

where L_{cls} is for category classification task, and cross entropy loss [5] and focal loss [6] are usually adopted. L_{reg} is for bounding box regression task and smooth L_1 loss [5], L_2 loss [4], [17], L_1 loss [18], IoU loss [19], GIoU loss [20], *etc.*, can be adopted. $\mathbb{I}_{\{c_i^*>0\}}$ is the indicator function, being 1 if $c_i^* > 0$ (foreground) and 0 otherwise (background). λ_{cls} and λ_{reg} are both the factors used for normalization and re-weighting simultaneously. The above procedure can be cascaded several times to further refine the predictions through

Fig. 2. Examples of our newly labeled images from the COCO `train` split. In each pair, the left one is the original annotation, and the right one is ours.

regarding the predicted boxes from the previous stage as the anchors for the current stage [5], [15].

During the inference phase, $\boldsymbol{p}_i$ and $\boldsymbol{t}_i$ are predicted for each anchor a_i, and non-maximum suppression (NMS) is applied as post-processing to remove the redundant predictions at last.

B. Analyses on the Impact of Noise

In most previous works, the location annotations in datasets for evaluation are, most commonly and tacitly, treated as accurate ground truth labels without considering the possible man-made errors, which means the regression target $\boldsymbol{t}_i^*$ in Equation (1) is considered to be accurate. Such conditions can be satisfied for some publicly available datasets thanks to the elaborate labeling effort [1], [2], [3], whereas they are usually not held for some private and self-collected datasets. If the location annotations are not accurate enough due to the limited budget, how much is the impact on the performance of modern detectors? To answer this question, experiments for both synthesized and real-world scenarios are conducted in this paper.

At the beginning of our analyses, we vastly labeled a subset of COCO `train` split, up to 12k images with relaxed label restrictions. The only request for annotators is that they should make it possible for others to retrieve the right object according to their labeled bounding box. Besides, we provide annotators with another indicator from COCO annotations to ensure our new annotations share almost the same mistaken and missing errors to COCO. In this way, the only difference between COCO annotations and ours is that the location annotations from COCO are relatively fine-grained and accurate while ours are coarse and noisy. Some examples are shown in Fig. 2. We further analyze the pattern of the man-made noise in the location annotations by considering COCO as the clean ones, as shown in Fig. 3. We mainly have three observations as follows: 1) the variance of the noise is roughly linear to the object scale; 2) relative errors (obtained by normalizing the absolute coordinate errors with respect to corresponding object widths or heights) are mostly centered around 0; 3) the correlation coefficients between relative errors of different boundary coordinates are at a relatively low level.

Despite that we have labeled such a noisy subset in the sense of real-world noisy conditions, its small scale (*i.e.*, only 10% of COCO dataset) and lack of ability to manually control the level of noise hold us back to deliver a deeper analysis. So we
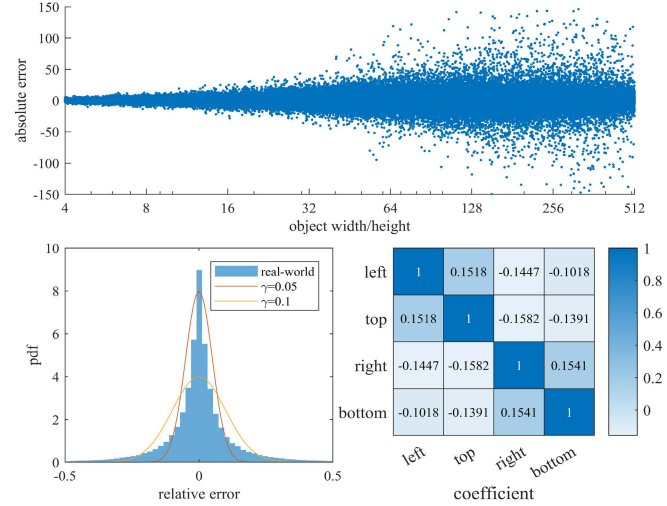

Fig. 3. Analyses of man-made noise in location annotations. Top: scatter diagram for absolute error of each boundary annotation with respect to corresponding object width or height. Bottom left: the distribution of relative error; the solid curves illustrate the distributions of our synthesized ones. Bottom right: correlation coefficients between relative errors of different boundary coordinates.

manually corrupt the location annotations by synthesized noise, following the above observations. Specifically, we add Gaussian noise to each box boundary coordinate separately for each instance in the `train` split. Take the left boundary (denoted as l^*) for example, the new coordinate could be $\tilde{l}^* = l^* + N_{l^*}$. The noise is set to be zero-mean with standard deviation linear to the object scale, *i.e.*, $N_{l^*} \sim \mathcal{N}(0, \sigma_{l^*}^2)$, where $\mathcal{N}$ denotes Gaussian distribution. We introduce the root mean squared relative error as the metric to measure the level of the noise, which is defined as, *e.g.*, $\gamma = \sqrt{\mathrm{E}[(N_{l^*}/w^*)^2]}$ for the left boundary. Thus, the level of the above Gaussian noise can be measured as $\gamma = \sigma_{l^*}/w^*$, where w^* is the object width. Illustrations are shown in Fig. 4. We let γ shared across four boundaries, *i.e.*, $\sigma_{r^*} = \gamma w^*$, $\sigma_{t^*} = \gamma h^*$, $\sigma_{b^*} = \gamma h^*$, where $\sigma_{r^*}, \sigma_{t^*}, \sigma_{b^*}$ are the standard deviations for the noise on the other three boundaries and h^* is the object height.

Based on our newly labeled and synthesized annotations on the COCO dataset, we take FCOS [9] and Faster R-CNN [5], two of the representative single-stage and two-stage detectors respectively, as examples to examine the impact of the noise.

TABLE I

NOTICEABLE DEGRADATION OF PERFORMANCE ON THE COCO MINIVAL SPLIT IS OBSERVED WHEN THE LOCATION
ANNOTATIONS OF THE TRAIN SPLIT ARE CORRUPTED BY SYNTHESIZED AND REAL-WORLD NOISE

| Methods	type	scale	γ	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
FCOS [9]	synthesized	123k	0	**38.5**	**57.3**	**41.6**	**22.5**	**42.4**	**49.7**
			0.05	37.1 (-1.4)	56.9	40.3	21.6	40.4	47.8
			0.1	33.5 (-5.3)	54.9	36.0	19.6	37.0	43.1
	real-world	12k	0	**22.8**	**39.2**	**23.3**	**11.6**	**24.8**	**30.0**
			-	21.2 (-1.6)	38.4	20.7	10.3	22.9	29.4
Faster R-CNN [5]	synthesized	123k	0	**37.5**	**58.2**	**40.9**	**21.1**	**41.2**	**49.0**
			0.05	35.8 (-1.7)	57.8	39.0	20.9	39.3	45.8
			0.1	33.3 (-4.2)	56.3	35.6	19.3	37.5	42.1
	real-world	12k	0	**23.3**	**42.9**	**22.7**	**11.7**	**25.8**	**30.8**
			-	21.8 (-1.5)	42.2	20.2	10.2	24.4	29.8
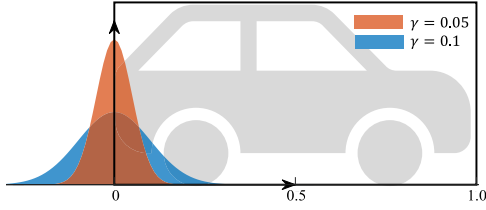
Fig. 4. Noise distributions we adopt for synthesized location annotations.
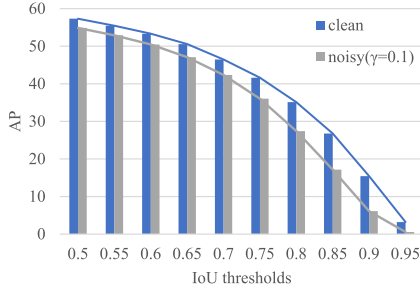
Fig. 5. APs under various IoU thresholds for the FCOS detectors trained with the clean and synthesized noisy annotations.

Fig. 6. Training curves for detectors with original and toy training losses.

FCOS is an anchor-free detector[2] with IoU-based regression loss, and the detector of Faster R-CNN utilizes *ROIAlign* operator to formulate a two-stage framework with L_p-norm-based regression loss. As shown in Fig. 1 and Table I, both the two methods are influenced by the injected noise. When $\gamma = 0.05$ in the case of synthesized noisy annotations, mAPs evaluated on the minival split degrade to 96.3% and 95.5% of the original scores for FCOS and Faster R-CNN respectively. We further increase γ to 0.1, and the performances decline to 87.0% and 88.8% of the original scores disastrously. Besides, we can find that APs with higher IoU thresholds drop heavier, as shown in Fig. 5, which indicates that the noise prominently damages the ability of detectors to accurately locate objects.

We also examine whether the detectors can overfit the noisy annotations. Since we have clean COCO annotations, it is easy and straightforward to monitor additional toy losses calculated by the predictions and the clean COCO annotations but stop the backward pass for computing gradients from these losses during the training phase. As shown in Fig. 6, we can find that the losses calculated by the noisy annotations are always higher than those calculated by the clean ones, and all of them

---

[2]We adopt the improved version with tricks like norm_on_bbox, centerness_on_reg, center_sampling and giou unless otherwise stated.
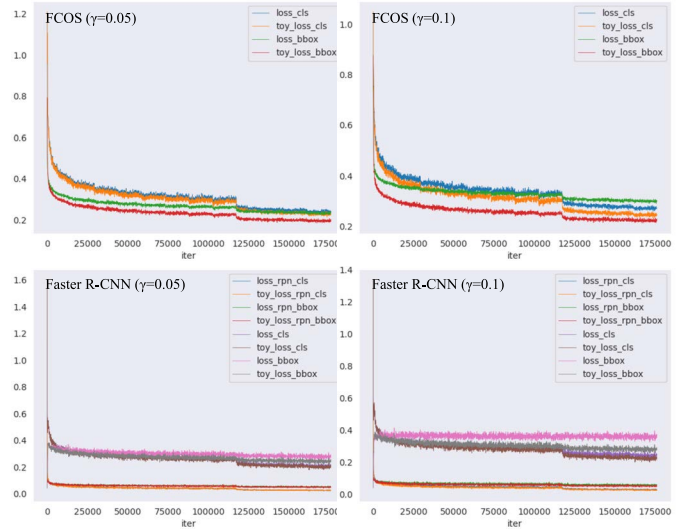
consistently show a downward trend during training. Thus we infer it is hard to overfit for detection tasks in our settings.

C. Our Improved Detector Training With Noise

When we happen to have a disappointing dataset with noisy location annotations due to the limited labor or time budget, it is thus worthy to study how to exploit these noisy annotations effectively as direct utilization of them results in prominent performance degradation as shown in Sec. III-B. In this section, a simple yet effective method is proposed based on the prediction ensemble to correct the noisy box annotations. Furthermore, to avoid the trivial solution during the direct use of this self-correction process in tracing, a Teacher-Student learning framework is adopted. Our method effectively alleviates the adverse effect of noisy annotations and achieves better performance on noisy datasets.

1) Prediction Ensemble: Due to the popular one-to-many label assignment during detector training [5], [6], [9], there are usually duplicate predictions corresponding to the same instance. We believe that these predictions possess the potential of correcting the noisy box annotations. To this end, we present the following problem: *For one object in the image with category label c^* and noisy box annotation $\boldsymbol{b}^*$, given multiple predictions $\mathcal{O} = \{O_1, O_2, \ldots, O_N\}$, how to acquire the corrected bounding box $\boldsymbol{b}^{cor}$ for that object?*

Here O_i is the prediction of the anchor a_i, and usually involves the classification score $\boldsymbol{p}_i$ and the box coordinate prediction $\boldsymbol{b}_i$. Note that $\boldsymbol{b}_i$ denotes the absolute coordinates in the image plane, which is the decoded box prediction transformed from $\boldsymbol{t}_i$, and N is the total number of anchors in the image.

The above problem can be formulated as the estimation of the posterior probability distribution, $p(\boldsymbol{b}^{cor}|c^*, \boldsymbol{b}^*, \mathcal{O})$. If we cast the coordinate prediction $\boldsymbol{b}_i$ in O_i as one measurement of $\boldsymbol{b}^{cor}$, and assume the measurements conditioned on $\boldsymbol{b}^{cor}$ are independently distributed, this problem can be considered as a particular case of recursive Bayesian estimation problem on *static systems*. If we further assume the measurement noise is normally distributed, it can be solved by Kalman filter [54]. In this *static system*, the output state vector for representing the corrected bounding box is updated by the above measurements that are introduced in a sequential fashion without considering their orders. This process is governed by the following equations:

$$
\begin{aligned}
\boldsymbol{b}_k^{cor} &= \boldsymbol{b}_{k-1}^{cor}, \\
\boldsymbol{b}_k &= \boldsymbol{b}_k^{cor} + \boldsymbol{v}_k,
\end{aligned}
\tag{2}
$$

where $p(\boldsymbol{v}_k) \sim \mathcal{N}(0, R_k)$ and R_k is the covariance matrix for the k-th measurement. This formulation can also be represented as $p(\boldsymbol{b}_k|\boldsymbol{b}_k^{cor}) \sim \mathcal{N}(\boldsymbol{b}_k^{cor}, R_k)$ in a probabilistic manner. The initial state can be modeled by the given noisy annotations, *i.e.*, $p(\boldsymbol{b}_0^{cor}) \sim \mathcal{N}(\boldsymbol{b}^*, P^*)$, where P^* is the covariance matrix for the annotations.

According to the derivations in Kalman filter [54], we can obtain the following recursive equations:

$$
\begin{aligned}
K_k &= P_{k-1}(P_{k-1} + R_k)^{-1}, \\
\boldsymbol{b}_k^{cor} &= \boldsymbol{b}_{k-1}^{cor} + K_k(\boldsymbol{b}_k - \boldsymbol{b}_{k-1}^{cor}), \\
P_k &= (I - K_k)P_{k-1},
\end{aligned}
\tag{3}
$$

where K_k is the Kalman gain and I is the identity matrix. Besides, we have the initial state as $\boldsymbol{b}_0^{cor} = \boldsymbol{b}^*$, $P_0 = P^*$.

By some derivations, the above recursive equations can be simplified to the following general terms, which can directly estimate the coordinates based on the noisy annotation and k predictions:

$$
\boldsymbol{b}_k^{cor} = P_k\left((P^*)^{-1}\boldsymbol{b}^* + \sum_{i=1}^{k} R_i^{-1}\boldsymbol{b}_i\right),
\tag{4}
$$

$$
P_k^{-1} = (P^*)^{-1} + \sum_{i=1}^{k} R_i^{-1}.
\tag{5}
$$

It is shown that the order of input sequence of predicted bounding boxes *does not* matter in estimating $\boldsymbol{b}_k^{cor}$ and P_k. More specifically, we can use all the N predictions to estimate the corrected coordinates as follows:

$$
\boldsymbol{b}^{cor} = \boldsymbol{b}_N^{cor} = (I + \sum_{i=1}^{N} P^* R_i^{-1})^{-1}(\boldsymbol{b}^* + \sum_{i=1}^{N} P^* R_i^{-1}\boldsymbol{b}_i^{cor}).
\tag{6}
$$

It can be drawn that $P^* R_i^{-1}$ matters in Equation (6) rather than any individual one of P^* and R_i. As the measurements
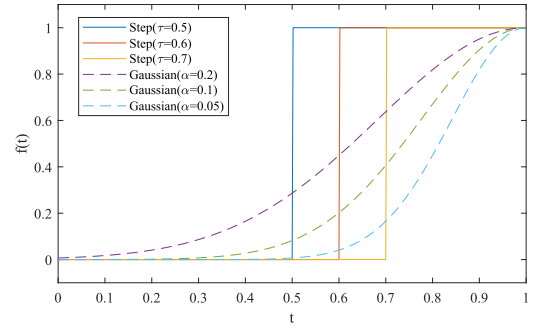

Fig. 7. Various choices for $f(t)$ in Equation (8): Step for $f(t) = \mathbb{I}(t \geqslant \tau)$ and Gaussian for $f(t) = e^{-\frac{(1-t)^2}{\alpha}}$.

(or predictions) are also trained with the supervision from the noisy annotations, we thus assume that the covariance matrix R_i for the i-th measurement inherits the correlationship in the covariance matrix P^* for the annotation and follows a scaling relation, *i.e.*, $P^* = \delta_i R_i$, where the scalar δ_i is the scale factor between P^* and R_i. Note that P^* and R_i are not necessarily to be diagonal matrices, which means that the four coordinates of the box are not necessarily independent to each other. Considering most objects in COCO [2] are subject to certain length/width ratios, they are indeed not independent. Following the assumption that $P^* = \delta_i R_i$, the Equation (6) can be rearranged to the one-dimensional case for each coordinate of the box. Here we take the estimation of the left boundary coordinate (l^{cor}) for example:

$$
l^{cor} = \frac{l^* + \sum_{i=1}^{N} \delta_i l_i}{1 + \sum_{i=1}^{N} \delta_i},
\tag{7}
$$

where l^* and l_i are the left boundary coordinate of the annotation and the i-th measurement.

Then, we discuss the design of δ_i. At least two factors should be considered: 1) the probability of the prediction's bounding box fitting the corresponding object; 2) the confidence of the prediction's category. To this end, we specially design δ_i as follows:

$$
\delta_i = f(IoU(\boldsymbol{b}_i, \boldsymbol{b}^*)) \cdot \boldsymbol{p}_{i,c^*}.
\tag{8}
$$

Here $\boldsymbol{p}_{i,c^*}$ represents the predicted classification score for category c^*, and $f(\cdot)$ should be an increasing function, and has several design choices. We can use step function ($f(t) = \mathbb{I}(t \geqslant \tau)$) or Gaussian function ($f(t) = e^{-\frac{(1-t)^2}{\alpha}}$) alternatively, where τ and α are the hyper-parameters as shown in Fig. 7. Both of them show comparably favorable performance in our experiments. The formulation in Equation (8) makes sense for the reason that: 1) higher IoU between the manually annotated bounding box and predicted one indicates a higher probability that they refer to the same object; 2) higher classification score for the corresponding category roughly indicates higher confidence for the bounding box prediction. Moreover, higher IoU and score together contribute to higher confidence of $\boldsymbol{b}_i$ and hence higher δ_i.

We will show the noise can be suppressed to a certain extent with the above prediction ensemble technique in Sec. IV-D.1.
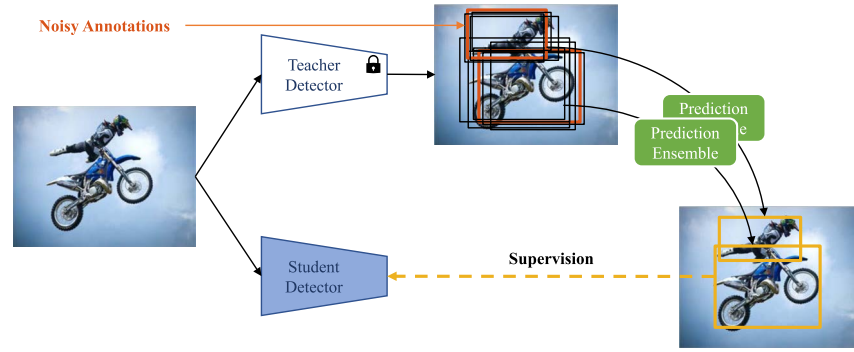
Fig. 8. The whole Teacher-Student learning framework. A frozen teacher detector and another student detector process images simultaneously. The predictions of the teacher detector are then utilized to correct the noisy bounding box annotations using our self-correction method described in III-C.1. The corrected annotations are used as supervision for the student detector.
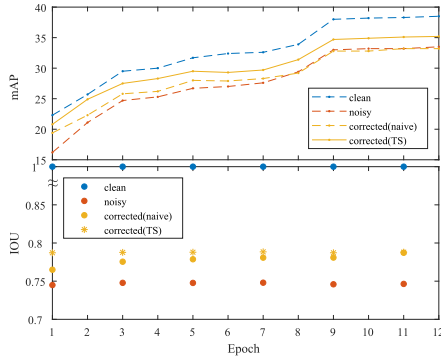

Fig. 9. In the training process, the top figure illustrates the mAP evaluated on `minival` split at each epoch for different training scenarios. The bottom one shows the average IoUs between the clean (noise-free) box annotations and the ones available in these training scenarios of training samples.

2) Teacher-Student Learning: Provided the above prediction ensemble method, a naive way to train a detector on the noisy dataset lies in applying such correcting method on the prediction obtained by the detector itself and utilizing the corrected annotations for supervision during training (denoted as `naive`). However, we experimentally find that it does not work to directly use this self-correction process in training. As shown in Fig. 9, although the corrected bounding boxes have higher IoUs than the clean ones than the original noisy annotations, the mAPs are not improved in the late of the process. We attribute it to the trivial solution obtained when the leading prediction dominates the ensemble. It is impracticable to count on the annotations corrected by the prediction itself to provide appropriate supervision.

To this end, a Teacher-Student learning framework is adopted in our proposed training process. Specifically, a teacher detector is introduced to correct the annotations, which is pre-trained on the noisy dataset and kept frozen during the training process of the student detector (denoted as `TS`). The overall framework is shown in Fig. 8. During training, the teacher and student detectors process the same images simultaneously. The predictions of the teacher detector are then utilized to correct the noisy bounding box annotations using our prediction ensemble method described in Sec. III-C.1. The corrected annotations are used as supervision for the student detector. In other words, the ability to alleviate noise of the teacher detector is transferred to the student detector, hence

boosting the student detector's locating performance with no additional latency during the inference phase.

IV. EXPERIMENTS

We present experimental results on the COCO [2] detection benchmark, which contains 118k images for training (`train` split), 5k images for validation (`minival` split) and 20k images for testing (`test-dev` split). We adopt the commonly used evaluation metric, AP, which is computed over ten different IoU thresholds, *i.e.*, 0.5:0.05:0.95. Considering the rather high quality of the bounding box annotations in COCO, we regard the original annotations as the clean ones without man-made noise. Thus, we investigate our methods by training the models on the noisy dataset with either our newly-labeled or synthesized annotations and evaluating them on the original COCO `minival` split or `test-dev` split.

A. Implementation Details

We benchmark on two representative detectors, FCOS [9] and Faster R-CNN [5], one of which is a single-stage method, and the other one follows the two-stage paradigm. All of our experiments are conducted based on mmdetection [18], a popular open-source codebase, and adopt the default settings, except that we train FCOS [9] and Faster R-CNN [5] with a batch size of 8 and halve the learning rate according to linear scaling rule [55], *i.e.*, 0.005 for FCOS and 0.01 for Faster R-CNN. Unless specified, the models are trained for 12 epochs, with the learning rate decreased by 0.1 at 8 and 11 epochs. We use ResNet-50 [56] as the default backbone and initialize it with the weights pre-trained on ImageNet [57]. FPN [7] is used by default. The input images are resized with the shorted side of 800 pixels and only horizontal image flipping is adopted as training data augmentations unless otherwise stated.

1) Training: We adopt the teacher-student learning framework described in Sec. III-C.2. First, a teacher model is obtained by training on the dataset with noisy annotations. Then, the student model is trained. During training, the forward process of the teacher model is executed synchronously and the predictions along with the noisy annotations are fed into our proposed prediction ensemble module to produce the new bounding boxes for objects, which are then used as the ground truths to generate the supervision of the student

TABLE II

EXPERIMENTAL RESULTS ON COCO `test-dev` SPLIT

Methods	type	scale	γ	Methods	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
FCOS [9]	synthesized	123k	0	-	38.9	57.9	42.0	22.4	41.4	48.6
			0.05	-	37.1	57.4	40.2	21.4	39.3	46.0
				Ours	**37.9 (+0.8)**	**58.1**	**41.3**	**22.1**	**40.6**	**47.0**
			0.1	-	33.6	55.4	36.0	19.6	35.8	41.4
				Ours	**35.6 (+2.0)**	**56.9**	**38.8**	**20.8**	**38.0**	**44.2**
	real-world	12k	0	-	22.9	39.4	23.6	11.8	24.2	28.5
			-	-	21.0	38.4	20.4	9.7	22.5	27.3
				Ours	**21.4 (+0.4)**	**38.8**	**21.0**	**10.2**	**23.0**	**27.4**
Faster R-CNN [5]	synthesized	123k	0	-	37.8	58.9	41.0	22.0	40.7	46.8
			0.05	-	36.4	58.5	**39.7**	**21.5**	39.4	44.5
				Ours	**36.7 (+0.3)**	**59.0**	39.4	21.2	**39.7**	**45.2**
			0.1	-	33.7	57.1	35.9	20.1	36.8	41.2
				Ours	**35.1 (+1.4)**	**58.1**	**37.6**	**20.4**	**38.2**	**43.2**
	real-world	12k	0	-	23.7	43.7	23.2	12.2	25.7	29.0
			-	-	22.2	42.7	20.6	10.5	24.5	28.2
				Ours	**22.5 (+0.3)**	**43.0**	**21.1**	**11.0**	**24.7**	**28.3**

detector. Following is a detailed description of how to get proper predictions from the teacher detector. For the FCOS, the predictions before NMS are utilized. Noting that the productions of classification scores and center-ness scores are considered as our category scores. As for the Faster R-CNN, the original one only produces sparse predictions after R-CNN, thus is not compatible with our methods. Nonetheless, a moderate modification is applied. Specifically, we collect the predictions of RPN and filter out the ones far away from any annotations (*e.g.*, IoU < 0.5) on the image plane or with rather low classification scores (*e.g.*, less than 0.05). All the remaining predicted boxes are fed into the R-CNN as ROIs to perform the second-stage classification and regression. To avoid bringing a heavy computing burden in some extreme cases, only top-k scored predictions from RPN are collected and k is set to 1000 in our experiments. Then all these predictions obtained by R-CNN are utilized to correct the noisy annotations. Noting that the above strategies can be easily and smoothly extended to other one-/two-stage detectors, though we only instantiate one detector for each kind of them for the demonstration purpose.

2) Inference: Only the student detector is required during the inference phase, thus no additional computational cost is brought by our method. All the configurations in this phase can follow the default settings without additional modifications. Specifically, in post-processing, for FCOS, we adopt NMS with an IoU threshold set to 0.6 and a score threshold of 0.05. For Faster R-CNN, NMS with a 0.7 IoU threshold is applied after RPN and only the top 1000 boxes according to their classification scores are fed into R-CNN to get the final predictions. Finally, it is followed by another NMS with an IoU threshold set to 0.5 and a score threshold of 0.05.

B. Overall Results

First, we evaluate our method on COCO `test-dev` split. All the models are trained on COCO `train` split with clean or noisy annotations. Different detectors (FCOS [9] and Faster R-CNN [5]) are adopted as our baselines and the performance under the noise of various levels is presented in Table II. It can be drawn that the performance is vulnerable to the noise in location annotations for both synthesized and real-world scenarios, while our method is capable to narrow the gap. Furthermore, more improvement *w.r.t.* APs of higher thresholds is achieved by our methods, which indicates that our method tends to help the detectors to locate objects precisely.

Besides, our method is only applied during the training phase, thus bringing no additional computational burdens for the inference phase. Considering only the forward pass is executed for the teacher detector and no intermediate results need to be cached, the runtime and memory occupation are also affected slightly in the training phase. Under our configuration, the training speed is about 80% of the original one, and the memory occupation is about 1.02 times. The negligible burden makes our method easy for utilization and deployment.

C. Comparisons With Existing Methods

In this section, we compare our method with some other existing methods. As stated in Sec. II-B, there are mainly two trends to deal with noisy labels. One focuses on reducing noisy samples' impact during training [12], [13], [22], [23], [24], [25], [26], [27], [28]. However, these methods are specially designed for classification tasks and there are no publicly available works for adapting them to detection tasks. Thus, we make an effort to adapt one representative method among them, *i.e.*, Co-teaching [28], to the detection task in our comparisons. Specifically, the two networks in the Co-teaching [28] framework are replaced with two detectors (here, we adopt Faster R-CNN), and they are trained simultaneously with a shared backbone. Following the common practice, each network samples its anchors with small classification loss as the relatively clean instances. The regression loss in its peer network is only calculated on these sampled anchors while the others are ignored.

The other trend to deal with noisy labels focuses on correcting the noisy category labels [10], [11], [29], [30] or noisy location annotations [39]. However, the method in [39] is only evaluated on the small Pascal VOC [1] dataset. Thus, we re-implement it under our setting as a strong counterpart.

The above two competitors are also trained on the COCO `train` split with our synthesized noisy annotations with $\gamma = 0.1$. All other configurations follow those in Sec. IV-A.

TABLE III
COMPARISONS WITH EXISTING METHODS

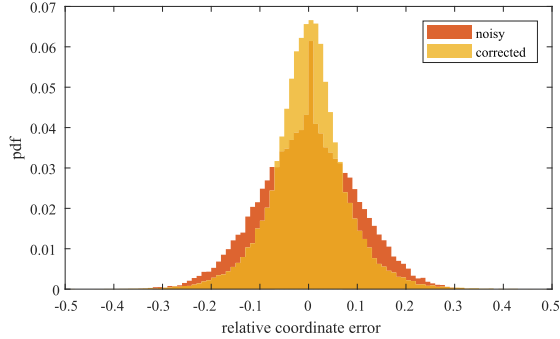| Method	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
-	33.3	56.3	35.6	19.3	37.5	42.1
Co-teaching [28]	33.8	57.0	36.1	20.6	37.8	42.8
Mao *et al.* [39]	33.6	55.0	36.0	19.7	37.0	43.1
ours	**34.6**	**57.2**	**36.9**	**21.8**	**38.2**	**43.4**

Fig. 10. Distributions of relative boundary coordinate errors for noisy annotations and our corrected ones.

The performance evaluated on the COCO `minival` split is presented in Table III. Our method achieves superior performance to the previous methods.

D. Ablation Study

In this section, we explore the effects of different components or design choices in our method. All the experiments are based on FCOS with synthesized Gaussian noise ($\gamma = 0.1$).

1) Prediction Ensemble: As stated in Sec. III-C.1, our prediction ensemble component can correct the noise in location annotations effectively. We plot the distributions of relative boundary coordinate errors for noisy annotations and our corrected ones in Fig. 10. The relative boundary coordinate errors are obtained by normalizing the absolute values of the difference between the noisy (or corrected) boundary coordinates and the accurate ones with respect to corresponding object widths or heights. Obviously, our corrected ones possess smaller variance, *i.e.*, lower location errors, and thus contribute to better detection performance.

Then, we examine different design choices of δ_i in Equation (8), the default of which is $\delta_i = f(IoU(\boldsymbol{b}_i, \boldsymbol{b}^*)) \cdot \boldsymbol{p}_{i,c^*}$. Without the former part related to IoU, we could not obtain meaningful results. It is anticipatable because numerous predictions not belonging to this object are involved in the ensemble and thus mislead the correcting procedure if the probability of the predictions corresponding to the object is excluded from consideration. In the case of removing the latter part ($\boldsymbol{p}_{i,c^*}$), slight performance degradation is observed, which confirms the effectiveness of using the scores. Moreover, we evaluate various functions for $f(\cdot)$, *e.g.*, step function and Gaussian function. We observe that both functions could achieve comparable performance with appropriate hyper-parameters provided. We infer that functions with similar tendencies are workable with fine-tuned hyper-parameters. The overall experimental results are listed in Table IV.

2) Teacher-Student Learning: In this section, we explore the essentials of our teacher-student learning framework.

TABLE IV
DIFFERENT DESIGN CHOICES OF δ_i IN EQUATION (8)

δ_i	param.	AP	AP_{50}	AP_{75}
$\boldsymbol{p}_{i,c^*}$	-	0.1	0.1	0.0
$\mathbb{I}(IoU(\boldsymbol{b}_i, \boldsymbol{b}^*) \geqslant \tau)$	$\tau = 0.7$	34.9	56.3	37.7
$\mathbb{I}(IoU(\boldsymbol{b}_i, \boldsymbol{b}^*) \geqslant \tau) \cdot \boldsymbol{p}_{i,c^*}$	$\tau = 0.5$	34.6	57.2	36.9
	$\tau = 0.6$	35.1	**56.9**	**37.9**
	$\tau = 0.7$	**35.2**	56.6	**37.9**
$e^{-\frac{(1-IoU(\boldsymbol{b}_i, \boldsymbol{b}^*))^2}{\alpha}} \cdot \boldsymbol{p}_{i,c^*}$	$\alpha = 0.2$	35.0	56.3	37.7
	$\alpha = 0.1$	**35.2**	**57.0**	**38.1**
	$\alpha = 0.05$	33.3	**57.0**	34.7

TABLE V
OUR METHOD ACHIEVES CONSISTENT IMPROVEMENT UNDER DIFFERENT
TRAINING CONFIGURATIONS. "R50" AND "R101" ARE SHORT FOR
RESNET50 AND RESNET101 [56] RESPECTIVELY. "MS" MEANS
TRAINING WITH MULTI-SCALE AUGMENTATION. "2×" IS
A TRAINING SCHEDULE THAT FOLLOWS THE SETTING
EXPLAINED IN MMDETECTION [18]

Config.	γ	Method	AP	AP_{50}	AP_{75}
R50+2×+MS	0	-	40.9	59.8	44.1
	0.05	-	38.8	58.6	42.2
		ours	**39.2**	**59.8**	**43.2**
	0.1	-	35.1	57.3	38.0
		ours	**36.2**	**57.9**	**39.3**
R101	0	-	40.7	59.6	43.9
	0.05	-	39.0	59.3	42.8
		ours	**39.8**	**59.9**	**43.5**
	0.1	-	35.7	58.1	38.3
		ours	**37.3**	**59.0**	**40.5**

As discussed in Sec. III-C.2, the trivial solution is easy to obtain if we rely on the predictions produced by the detector itself to serve as the source of the ensemble for supervision. As shown in Fig. 9, for this `naive` approach, we can find that the quality of the annotations has consistent improvement over the original noisy annotations thanks to our self-correction method. Even at the very beginning of the training process, the quality is improved, which we attribute to the dynamic filtering ability of our designed δ_i (Equation (8)). Great promotion on mAP is also witnessed for the first several epochs during training. However, the performance of the detector shows no further improvement over the one without correction for the last several epochs, though the IoU is still increasing. We may attribute it to that such an ensemble could not provide sufficient valuable correction for supervision in this stage.

In our teacher-student learning framework (`TS`), the predictions used for correction and the ones under supervision are decoupled, and thus the trivial solution is avoided naturally. As shown in Fig. 9, the performance of the detectors is consistently improved during our whole training process.

E. Extensive Experiments

1) Other Training Configurations: We examine the results of our method under other training configurations in this section as shown in Table V. First, we test a longer training schedule with multi-scale training augmentation (R50+2×+MS). Note that the configurations are adopted for both teacher and student training in our method. It can be found that even though better training configurations are adopted, the performance is still inferior when the location annotations are corrupted. But the gap can be narrowed by using our method as ever. A similar phenomenon is observed in the case of a larger backbone (R101).

TABLE VI

TRAINING WITH THE DECOUPLED TEACHER AND STUDENT CONFIGURA-
TIONS. THE ROWS WITH "S CONFIG." MARKED AS "−" SHOW THE
RESULTS OF ISOLATED TEACHER DETECTORS. THE ROWS WITH "T
CONFIG." MARKED AS "−" PRESENT THE RESULTS OBTAINED
WITHOUT OUR CORRECTION METHOD AND TEACHER-
STUDENT LEARNING, WHICH ARE ALSO OUR BASE-
LINES FOR EACH GROUP. THE ROWS WITH "T CON-
FIG." MARKED AS "R50*" ARE FOR THE COM-
PARISON WITH THE ITERATIVE TRAINING
STRATEGY. BETTER PERFORMANCE IS
ACHIEVED WHEN STRONGER TEACH-
ERS ARE ADOPTED

S Config.	T Config.	γ	AP
-	R50+2×+MS	0	40.9
-	R101	0	40.7
R50	-	0	38.5
	-		37.1
	R50		37.7 (+0.6)
	R50*	0.05	37.5 (+0.4)
	R50+2×+MS		37.8 (+0.7)
	R101		**37.9 (+0.8)**
	-		33.5
	R50		35.2 (+1.7)
	R50*	0.1	35.1 (+1.6)
	R50+2×+MS		35.4 (+1.9)
	R101		**35.5 (+2.0)**

TABLE VII

ROBUSTNESS TEST WITH 3 RUNS FOR EACH CONFIGURATION

	Config	mAP		
		γ=0	γ=0.05	γ=0.1
a.	Same annotations without noise	38.5(±0.0)	-	-
b.	annotations with the same noise	-	37.1(±0.1)	33.7(±0.2)
c.	+ our methods	-	37.7(±0.1)	35.3(±0.1)
d.	annotations with different noise	-	37.1(±0.2)	33.4(±0.2)
e.	+ our methods	-	37.7(±0.1)	35.1(±0.1)

Then we test the case with the training configurations of the teacher and student detectors decoupled, *e.g.*, a stronger teacher is obtained first and then utilized to correct the supervision of the student detector. The experimental results are shown in Table VI. First, we compare with the iterative training strategy, *i.e.*, the student detector obtained from our method is then used as a teacher for the second round of training in our framework. However, we observe no continuous improvement, which we attribute to too much inductive bias involved in iterative training, leading to accumulated errors. Then we test the stronger teachers obtained by the longer training schedule with multi-scale training augmentation (R50+2×+MS) or a larger backbone (R101). We observe that stronger teachers lead to better student performance, but the additional improvement is limited. An upper bound is likely to exist, which prevents the performance from approximating to the optimal result infinitely.

2) Robustness: As randomness is involved in our noise generation, the robustness of our method should be taken into account. So we run 3 times for each configuration to test it, as shown in Table VII. First, we conduct experiments on datasets with the same clean (a) or noisy (b and c) annotations for each run. In this case, the randomness comes from the random initialization of the model weights and the random sampling order of the data loader. We find that the performance varies much more if the annotations are noisy (compare b to a). Then, we execute noise generation 3 times resulting in 3 different sets of noisy annotations for the dataset

TABLE VIII

EXPERIMENTAL RESULTS FOR DETECTORS TRAINED WITH
OUR NEWLY LABELED SUBSET OF COCO

dataset & methods	scale	time	AP	AP_{50}	AP_{75}
COCO labels	12k	-	22.8	39.2	23.3
our noisy labels	12k	364h	21.2	38.4	20.7
+ our method			**21.5**	**38.9**	**21.2**
elaborate labels	8k	372h	19.8	35.0	19.8

TABLE IX

ASYMMETRICAL NOISE HAS RATHER MORE TERRIBLE IMPACT ON THE
DETECTORS' PERFORMANCE THAN THE SYMMETRICAL ONES

Noise	Method	AP	AP_{50}	AP_{75}
-	-	38.5	57.3	41.6
Sym. ($\gamma = 0.05$)	-	37.1	56.9	40.6
Asym.	-	34.6	56.6	37.9
($\gamma = 0.05$, enclosing)	ours	35.1	56.6	38.8
Asym.	-	34.1	56.4	37.4
($\gamma = 0.05$, enclosed)	ours	34.5	56.4	38.0

(d and e). This time, the performance possesses a larger variance due to the additional source of randomness, *i.e.*, noise in the annotations (compare d to b). Furthermore, our method achieves consistent gains across different settings (compare c to b and e to d), which demonstrates the robustness of our method.

F. Label Time Analysis

In this section, we analyze the time consumption of labeling the images. It costs us about 364 human hours to label 12k images with rough bounding box annotations. Nevertheless, we can only label about 8k images with fine-grained bounding box annotations by taking up nearly the same time. That is because much more time is consumed to finely adjust the boundary annotations. Thus, about 1.5 times faster labeling speed can be achieved by only requiring rough bounding box annotation. We believe that much labeling time can be saved if we further loosen the label restrictions. With the 8k images with elaborate labels, we can train a detector as shown in Table VIII. Its performance is far behind those trained with 12k images, especially for AP_{50} (up to −4.2 gap), though they are labeled with almost equal time consumed. While by leveraging our method, we can narrow the gap for AP_{50} to only −0.3. We may conclude that, with a limited or restricted label budget, it is preferable to label with rough box annotations rather than elaborate but fewer ones.

G. Evaluation on Other Forms of Noise

Finally, we introduce two additional forms of noisy location annotations for analyzing their impact on the performance degradation. We believe this can further provide positive guidance for data annotation. Specifically, we adopt asymmetrical noise, one of which restricts the noisy annotation box to completely enclose the ideal one and the other completely enclosed by the ideal one. Exponential distributions with hyper-parameter λ are adopted. We set $\lambda = 20\sqrt{2}$, and thus the noise level is $\gamma = 0.05$. We find that the asymmetrical noise has a rather more severe impact on the detectors' performance degradation than the symmetrical one (Gaussian distribution) as shown in Table IX, with APs of higher IoU thresholds

dropping devastatingly. While our method can still reduce the negative impact to a certain extent. Based on the above observations, we conclude that it is better to control the distribution of the annotated coordinates centered as near as possible to the ideal one, and neither consistently larger nor smaller noisy annotations than the ideal one are better choices for box labeling.

V. CONCLUSION

The impact of noisy location annotations for detection tasks is investigated in the paper. We demonstrate that fine-grained location annotations are crucial for the plausible performance of modern detectors, especially for the AP performance with high IoU thresholds. Then a simple yet effective method is proposed to correct the noise in the location annotations, which can remarkably improve the performance of detectors under both synthesized noisy annotations and real-world man-made noisy annotations. Besides, we find that if there is a limited or restricted label budget, the rough location annotations with some weak noise can significantly reduce label cost, and utilizing our method is essential for performance improvement. Furthermore, in our view, more studies should be focused on the pattern of man-made noise, based on which much more effective methods could be proposed and further improve the detector performance. Finally, the combination of location noise and category label noise is also a challenging topic, which needs future exploration.

ACKNOWLEDGMENT

REFERENCES

[1] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[2] T. Y. Lin, M. Maire, S. Belongie, J. Hays, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[3] S. Shao et al., "Objects365: A large-scale, high-quality dataset for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8430–8439.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 779–788.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[6] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[7] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

[8] W. Liu et al., "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[9] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9627–9636.

[10] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, "Training deep neural networks on noisy labels with bootstrapping," 2014, *arXiv:1412.6596*.

[11] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, "Learning from massive noisy labeled data for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2691–2699.

[12] Z. Zhang and M. R. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Proc. 32nd Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 1–11.

[13] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, "MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2304–2313.

[14] S. Verbaeten and A. Van Assche, "Ensemble methods for noise elimination in classification problems," in *Multiple Classifier Systems*, T. Windeatt and F. Roli, Eds. Berlin, Germany: Springer, 2003, pp. 317–325.

[15] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6154–6162.

[16] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019, *arXiv:1904.07850*.

[17] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[18] K. Chen et al., "MMDetection: Open MMLab detection toolbox and benchmark," 2019, *arXiv:1906.07155*.

[19] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "UnitBox: An advanced object detection network," in *Proc. 24th ACM Int. Conf. Multimedia*. New York, NY, USA: Association for Computing Machinery, Oct. 2016, pp. 516–520.

[20] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 658–666.

[21] Y. He, C. Zhu, J. Wang, M. Savvides, and X. Zhang, "Bounding box regression with uncertainty for accurate object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2888–2897.

[22] E. Arazo, D. Ortego, P. Albert, N. O'Connor, and K. McGuinness, "Unsupervised label noise modeling and loss correction," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 312–321.

[23] A. Ghosh, H. Kumar, and P. Sastry, "Robust loss functions under label noise for deep neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2017, vol. 31, no. 1, pp. 1–7.

[24] J. Li, Y. Wong, Q. Zhao, and M. S. Kankanhalli, "Learning to learn from noisy labeled data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5051–5059.

[25] A. K. Menon, A. S. Rawat, S. J. Reddi, and S. Kumar, "Can gradient clipping mitigate label noise?" in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–26. [Online]. Available: https://openreview.net/forum?id=rklB76EKPr

[26] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4334–4343.

[27] E. Malach and S. Shalev-Shwartz, "Decoupling 'when to update' from 'how to update,'" in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[28] B. Han et al., "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–11.

[29] D. Tanaka, D. Ikami, T. Yamasaki, and K. Aizawa, "Joint optimization framework for learning with noisy labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5552–5560.

[30] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and L.-J. Li, "Learning from noisy labels with distillation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1910–1918.

[31] A. Vahdat, "Toward robustness against label noise in training deep discriminative neural networks," 2017, *arXiv:1706.00038*.

[32] J. Jeong, S. Lee, J. Kim, and N. Kwak, "Consistency-based semi-supervised learning for object detection," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 10759–10768.

[33] K. Sohn, Z. Zhang, C.-L. Li, H. Zhang, C.-Y. Lee, and T. Pfister, "A simple semi-supervised learning framework for object detection," 2020, *arXiv:2005.04757*.

[34] Y.-C. Liu et al., "Unbiased teacher for semi-supervised object detection," 2021, *arXiv:2102.09480*.

[35] H. Bilen and A. Vedaldi, "Weakly supervised deep detection networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2846–2854.

[36] P. Tang, X. Wang, X. Bai, and W. Liu, "Multiple instance detection network with online instance classifier refinement," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2843–2851.

[37] P. Tang et al., "Weakly supervised region proposal network and object detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 352–368.

[38] C. Rosenberg, M. Hebert, and H. Schneiderman, "Semi-supervised self-training of object detection models," in *Proc. IEEE Workshops Appl. Comput. Vis.*, Jan. 2005, pp. 29–36.

[39] J. Mao, Q. Yu, and K. Aizawa, "Noisy localization annotation refinement for object detection," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 2006–2010.

[40] S. Chadwick and P. Newman, "Training object detectors with noisy data," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 1319–1325.

[41] J. Mao, Q. Yu, Y. Yamakata, and K. Aizawa, "Noisy annotation refinement for object detection," *The Brit. Mach. Vis. Conf. (BMVC)*, 2021, pp. 1–12.

[42] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[43] B. Heo, J. Kim, S. Yun, H. Park, N. Kwak, and J. Y. Choi, "A comprehensive overhaul of feature distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1921–1930.

[44] F. Tung and G. Mori, "Similarity-preserving knowledge distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1365–1374.

[45] S. You, C. Xu, C. Xu, and D. Tao, "Learning from multiple teacher networks," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 1285–1294.

[46] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 742–751.

[47] Q. Li, S. Jin, and J. Yan, "Mimicking very efficient network for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6356–6364.

[48] T. He, C. Shen, Z. Tian, D. Gong, C. Sun, and Y. Yan, "Knowledge adaptation for efficient semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 578–587.

[49] Y. Liu, K. Chen, C. Liu, Z. Qin, Z. Luo, and J. Wang, "Structured knowledge distillation for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2604–2613.

[50] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," 2017, *arXiv:1703.01780*.

[51] K. Sohn et al., "FixMatch: Simplifying semi-supervised learning with consistency and confidence," 2020, *arXiv:2001.07685*.

[52] M. Caron et al., "Emerging properties in self-supervised vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9650–9660.

[53] Z. Meng, J. Li, Y. Gong, and B.-H. Juang, "Adversarial teacher-student learning for unsupervised domain adaptation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 5949–5953.

[54] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, J. Basic Eng.*, vol. 82, no.1, pp. 35–45, May 1960.

[55] P. Goyal et al., "Accurate, large minibatch SGD: Training ImageNet in 1 hour," 2017, *arXiv:1706.02677*.

[56] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[57] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. CVPR*, Jun. 2009, pp. 248–255.

Shaoru Wang received the B.E. degree from the Department of the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China, in 2018. He is currently pursuing the Ph.D. degree with the Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing.

Jin Gao received the B.S. degree from Beihang University, Beijing, China, in 2010, and the Ph.D. degree from the University of Chinese Academy of Sciences (UCAS) in 2015. He is currently an Associate Professor with the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA). His research interests include visual tracking, autonomous vehicles, and service robots.

Bing Li received the Ph.D. degree in computer science and engineering from Beijing Jiaotong University, Beijing, China, in 2009. He is currently an Associate Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing. His research interests include color constancy, visual saliency, and web content mining.

Weiming Hu (Senior Member, IEEE) received the Ph.D. degree from the Department of Computer Science and Engineering, Zhejiang University, Hangzhou, China, in 1998. From 1998 to 2000, he was a Postdoctoral Research Fellow with the Institute of Computer Science and Technology, Peking University, Beijing. He is currently a Professor with the Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing. His research interests include visual motion analysis, recognition of web objectionable information, and network intrusion detection.