α-IoU: A Family of Power Intersection over Union Losses for Bounding Box Regression

Anonymous Author(s)
Affiliation
Address
email

Abstract

Bounding box (bbox) regression is a fundamental task in computer vision. So far, the most commonly used loss functions for bbox regression are the Intersection over Union (IoU) loss and its variants. In this paper, we generalize existing IoU-based losses to a new family of power IoU losses that have a power IoU term and an additional power regularization term with a single power parameter α. We call this new family of losses the α-IoU losses and analyze properties such as order preservingness and loss/gradient reweighting. Experiments on multiple object detection benchmarks and models demonstrate that α-IoU losses, 1) can surpass existing IoU-based losses by a noticeable performance margin; 2) offer detectors more flexibility in achieving different levels of bbox regression accuracy by modulating α; and 3) are more robust to small datasets and noisy bboxes.

1 Introduction

Bounding box (bbox) regression localizes an object in an image/video by predicting a bbox for the object, which is fundamental to object detection, localization, and tracking. For example, the most advanced object detectors often consist of a bbox regression branch and a classification branch with the bbox regression branch generating bboxes to localize objects for classification. In this work, we explore more effective loss functions for bbox regression in the context of object detection.

Whilst early works in object detection use ℓ_n-norm losses [11] for bbox regression, recent works directly adopt the localization performance metric, i.e., Intersection over Union (IoU), as the localization loss [24, 34]. Compared with ℓ_n-norm losses, the IoU loss is invariant to bbox scales, thus helping train better detectors. However, the IoU loss suffers from the gradient vanishing problem when predicted bboxes are not overlapping with the ground truth. This tends to slow down convergence and result in inaccurate detectors, which motivates the design of several improved IoU-based losses including Generalized IoU (GIoU), Distance-IoU (DIoU) and Complete IoU (CIoU). GIoU introduces a penalty term into the IoU loss to alleviate the gradient vanishing problem [28], while DIoU and CIoU consider the central point distance and aspect ratio between predicted bboxes and their ground truth in penalty terms [38].

In this paper, we present a new family of IoU losses obtained by applying power transformations to existing IoU-based losses. We first apply the Box-Cox transformation [2] to the IoU loss $\mathcal{L}_{\text{IoU}} = 1 - IoU$ and generalize it to a power IoU loss: $\mathcal{L}_{\alpha\text{-IoU}} = (1 - IoU^{\alpha})/\alpha,\ \alpha > 0$, denoted as α-IoU. We further simplify α-IoU to $\mathcal{L}_{\alpha\text{-IoU}} = 1 - IoU^{\alpha}$ for $\alpha \nrightarrow 0$ and extend it to a more general form with an additional power regularization term (see equation (3)). This allows us to generalize existing IoU-based losses, including GIoU, DIoU, and CIoU, to a new family of power IoU losses (see equation (4)) for more accurate bbox regression as well as object detection.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

We show that, relative to $\mathcal{L}_{\text{IoU}}$, $\mathcal{L}_{\alpha\text{-IoU}}$ with $\alpha > 1$ up-weights both the loss and gradient of high IoU objects, leading to improved bbox regression accuracy. When $0 < \alpha < 1$, it down-weights high IoU objects which we find hurts regression accuracy. The power parameter α can serve as a knob to adapt α-IoU losses to meeting different levels of bbox regression accuracy (precision measured under different IoU thresholds), with $\alpha > 1$ for high regression accuracy (i.e., high IoU thresholds) by focusing more on those high IoU objects. We also empirically show that α is not sensitive to different models or datasets, with $\alpha = 3$ performing consistently well in most cases. The family of α-IoU losses can be easily applied for improving state-of-the-art detectors under both clean and noisy bbox settings without introducing additional parameters to these models, making any modifications to training algorithms, nor decreasing the inference speed (FPS).

In summary, our main contributions are as follows:

- We propose a new family of power IoU losses called α-IoU for accurate bbox regression and object detection. α-IoU presents a unified power generalization of existing IoU-based losses.
- We analyze a set of properties of α-IoU, including order preservingness and loss/gradient reweighting, to show that a proper choice of α (i.e., $\alpha > 1$) can help improve bbox regression accuracy by adaptively up-weighting the loss and gradient of high IoU objects.
- We empirically show, on multiple benchmark object detection datasets and models, that α-IoU losses can consistently outperform existing IoU-based losses and provide more robustness for small datasets and noisy bboxes.

2 Related Work

Object Detection Models. There exist two mainstream types of detection models: anchor-based and anchor-free detectors. Anchor-based detectors can be further divided into two-stage and one-stage models. Two-stage anchor-based detectors (e.g., R-CNN series [11, 27, 12, 3], HTC [5], and TSD [29]) are firstly proposed in object detection tasks, which are composed of region proposal networks (RPNs) and classifiers. RPNs generate a large number of foreground and background region proposals, followed by networks to classify objects in the proposals. Towards real-time object detection, one-stage anchor-based detectors (e.g., YOLO series [25, 26, 1], RetinaNet [19], and SSD [22]) are developed to predict bboxes and categories at the same time, thus no longer need RPNs. Anchor boxes with prior scales and aspect ratios should be defined before training anchor-based detectors. Techniques have been proposed to mitigate the sensitivity of these models to hand-picked anchor boxes, for example, attention-based fusion networks [27] and clustering algorithms [26]. These techniques learn prior anchors from the training set for every sliding window or grid cell.

Recently, anchor-free detectors such as CornerNet [14], CenterNet$_1$ [8], ExtremeNet [40], and CentripetalNet [7], have also been proposed to get rid of anchor priors. These models first predict locations of keypoints (corners, centroids, or extreme points), then group them into same bboxes if they are geometrically aligned. There also exist other models that generate pixel-wise results. For example, CenterNet$_2$ estimates pixel-level categories of objects along with their sizes and offsets [39]. FCOS generates pixel-wise classification, centerness, and bbox (top, down, left, right) results using multi-head CNNs [30], followed by the Adaptive Training Sample Selection (ATSS) [35] as an improvement on automatically selecting positive and negative samples. In addition, transformers (e.g., DETR series [4, 41]) have also been developed for object detection without anchor generation or non-maximum suppression (NMS), achieving the performance on par with above CNN-based detectors. In this work, we propose a new family of generalized IoU losses to improve the performance of these detectors without making any architectural modifications, which is orthogonal to the above research.

Bounding Box Regression Losses. Anchor-based detectors regress offsets between ground-truth bboxes and their closest anchors, while anchor-free detectors predict keypoints of objects with some frameworks also generating sizes of bboxes. These predicted offsets or keypoints (w/ or w/o bbox sizes) are then mapped back to the pixel space for generating bboxes. Localization losses usually compare generated bboxes with their ground truth. Early works adopt ℓ_n-norm losses [11] for bbox regression, which have been found sensitive to varying bbox scales. Recent works replace them with the IoU loss and its variants such as BIoU, GIoU, DIoU and CIoU for bbox regression, as IoU is the metric for localization and scale-invariant [24, 34]. The Bounded IoU (BIoU) loss maximizes the IoU overlap between the region of interest (RoI) and the ground truth based on a set of IoU upper bounds [31]. GIoU is proposed to address the problem of gradient vanishing on non-overlapping

examples, which are examples having non-overlapping predicted bboxes with the ground truth (IoU is zero) [28]. DIoU and CIoU [38] losses further consider the overlapping area, central point distance, and aspect ratio in IoU and regularization terms. These regularization terms can help improve the convergence speed as well as the final detection performance. There are also losses designed to focus more on high IoU objects, for example, the Rectified IoU (RIoU) loss [32], and the Focal and Efficient IoU (Focal-EIoU) loss [36]. These loss functions increase gradients of those examples that are in high bbox regression accuracy. However, RIoU and Focal-EIoU are neither concise nor generalized compared with other IoU-based losses. In this paper, we apply a power transformation to generalize the above vanilla IoU loss and regularized IoU-based losses for both their IoU and regularization terms. The new family of generalized losses improve bbox regression accuracy by adaptively reweighting the loss and gradient of high and low IoU objects.

There are also works on AutoML-based loss function search for computer vision tasks [21, 16, 15]. Despite their advantage in saving human efforts, these methods are very expensive in searching qualified loss functions (e.g., days of searching time on multiple GPUs), and probably with limited performance improvement based on existing losses [21]. We will empirically compare with one of these losses in following experiments.

3 α-IoU Losses for Bounding Box Regression

3.1 Preliminaries

We study the problem of bbox regression in object detection. Let $\boldsymbol{X} \in \mathcal{R}^{d_x}$ be the input space and $\boldsymbol{Y} \in \mathcal{R}^{d_y}$ be the annotation space, with d_x and d_y denoting the input and annotation dimensions, respectively. Given a dataset $D = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{n}$ of n training examples with each $(\boldsymbol{x}_i, \boldsymbol{y}_i) \in (\boldsymbol{X} \times \boldsymbol{Y})$, the task is to learn a function f (represented by a detector network) that maps the input space to the annotation space $f : \boldsymbol{X} \to \boldsymbol{Y}$. In object detection, each $\boldsymbol{y}_i = (c_{i,k}, B_{i,k})_{k=1}^{m_i}$, where m_i is the total number of objects in $\boldsymbol{x}_i$, $c_{i,k}$ is the category of the kth object in $\boldsymbol{x}_i$ and $B_{i,k}$ is its bbox.

The bbox regression performance is measured by the Intersection over Union (IoU) metric between the predicted bbox B and the ground truth B^{gt}: $IoU = |B \cap B^{gt}| / |B \cup B^{gt}|$. Positive examples (both true and false positives) are determined from the set of predictions according to an IoU threshold, based on which the Average Precision (AP) over all categories of objects can be calculated. E.g., AP_{50} measures the AP of objects localized by bboxes with an IoU that is above the threshold 0.5. The final performance of a detector is commonly evaluated by the mean Average Precision (mAP) across multiple IoU thresholds. For instance, the popular metric $\text{mAP}_{50:95}$ measures the mAP of examples across the set of IoU thresholds ranging from 0.5 to 0.95 with a stride of 0.05.

3.2 α-IoU Losses

The vanilla IoU loss is defined as $\mathcal{L}_{\text{IoU}} = 1 - IoU$. We first apply the Box-Cox transformation[1] [2] and generalize the IoU loss to an α-IoU loss:

$$\mathcal{L}_{\alpha\text{-IoU}} = \frac{1 - IoU^{\alpha}}{\alpha}, \alpha > 0. \tag{1}$$

By modulating the parameter α in α-IoU, one can derive most of the IoU terms in existing losses, e.g., $\log(IoU)$, IoU and IoU^2. When $\alpha \to 0$, we obtain $\lim_{\alpha \to 0} \mathcal{L}_{\alpha\text{-IoU}} = -\log(IoU) = \mathcal{L}_{\text{log(IoU)}}$ [34] (see the proof in Appendix A). We recover the IoU loss with $\alpha = 1$: $\mathcal{L}_{\text{1-IoU}} = 1 - IoU = \mathcal{L}_{\text{IoU}}$. And $\mathcal{L}_{\text{2-IoU}} = \frac{1}{2}(1 - IoU^2) = \frac{1}{2}\mathcal{L}_{\text{IoU}^2}$, when $\alpha = 2$. We can also extend the above α-IoU formula to loss functions with multiple IoU terms (e.g. RIoU [32]) by using multiple α values.

We simplify the above α-IoU formula for $\alpha > 0$ and $\alpha \nrightarrow 0$, as in this case, the denominator α in equation (1) is just a positive constant and will not affect minimization of the objective. This gives us

[1] The Box-Cox transformation has also been applied for generalizing the Cross Entropy (CE) loss into the Generalized Cross Entropy (GCE) loss [37]. Such a generalization unifies the Mean Absolute Error (MAE) and the CE loss, increasing the robustness to noisy labels for classification tasks. However, GCE may slow down convergence and lead to degraded performance [33, 23]. In contrast, our α-IoU generalization is for bbox regression and can improve the localization performance on both clean and noisy bbox datasets.

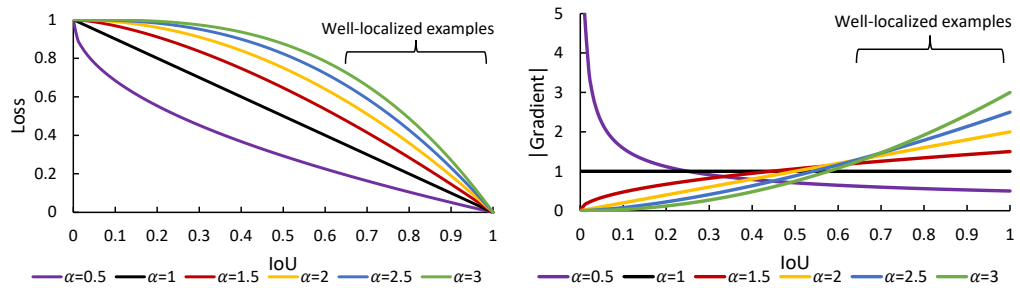

Figure 1: Correlation between IoU and $\mathcal{L}_{\alpha\text{-IoU}} = 1 - IoU^{\alpha}$ (left) and its absolute gradient $|\nabla_{\text{IoU}}\mathcal{L}_{\alpha\text{-IoU}}|$ (right) with different $\alpha \in [0.5, 3]$. According to both plots, $\mathcal{L}_{\alpha\text{-IoU}}$ reweights all objects adaptively and distinctively for $0 < \alpha < 1$ vs. $\alpha > 1$ ($\alpha = 1$ marks the IoU loss).

two cases of the α-IoU loss for $\alpha \to 0$ and $\alpha \nrightarrow 0$, respectively:

$$\mathcal{L}_{\alpha\text{-IoU}} = \begin{cases} -\log(IoU), & \alpha \to 0, \\ 1 - IoU^{\alpha}, & \alpha \nrightarrow 0. \end{cases} \tag{2}$$

Here, we are more interested in the case $\alpha \nrightarrow 0$ as most state-of-the-art IoU-based losses have an $\alpha \geq 1$. We then extend the above α-IoU loss for $\alpha \nrightarrow 0$ to a more general form by introducing a power penalty/regularization term into the formula:

$$\mathcal{L}_{\alpha\text{-IoU}} = 1 - IoU^{\alpha_1} + \mathcal{P}^{\alpha_2}(B, B^{gt}), \tag{3}$$

where $\alpha_1 > 0$, $\alpha_2 > 0$, and $\mathcal{P}^{\alpha_2}(B, B^{gt})$ denotes any penalty term computed based on B and B^{gt}. This simple extension allows straightforward generalization of existing IoU-based losses to their α-IoU versions. In Appendix B.2.1, We empirically show that maintaining the power consistency between the IoU term and the penalty term (i.e., $\alpha_1 = \alpha_2$) helps train better detectors.

With the above α-IoU formula, we generalize four commonly used IoU-based losses including $\mathcal{L}_{\text{IoU}}$, $\mathcal{L}_{\text{GIoU}}$, $\mathcal{L}_{\text{DIoU}}$, and $\mathcal{L}_{\text{CIoU}}$ using the same power parameter α for IoU terms and penalty terms:

$$\begin{aligned} \mathcal{L}_{\text{IoU}} &= 1 - IoU \Longrightarrow \mathcal{L}_{\alpha\text{-IoU}} = 1 - IoU^{\alpha}, \\ \mathcal{L}_{\text{GIoU}} &= 1 - IoU + \frac{|C \setminus (B \cup B^{gt})|}{|C|} \Longrightarrow \mathcal{L}_{\alpha\text{-GIoU}} = 1 - IoU^{\alpha} + (\frac{|C \setminus (B \cup B^{gt})|}{|C|})^{\alpha}, \\ \mathcal{L}_{\text{DIoU}} &= 1 - IoU + \frac{\rho^2(\boldsymbol{b}, \boldsymbol{b}^{gt})}{c^2} \Longrightarrow \mathcal{L}_{\alpha\text{-DIoU}} = 1 - IoU^{\alpha} + \frac{\rho^{2\alpha}(\boldsymbol{b}, \boldsymbol{b}^{gt})}{c^{2\alpha}}, \\ \mathcal{L}_{\text{CIoU}} &= 1 - IoU + \frac{\rho^2(\boldsymbol{b}, \boldsymbol{b}^{gt})}{c^2} + \beta v \Longrightarrow \mathcal{L}_{\alpha\text{-CIoU}} = 1 - IoU^{\alpha} + \frac{\rho^{2\alpha}(\boldsymbol{b}, \boldsymbol{b}^{gt})}{c^{2\alpha}} + (\beta v)^{\alpha}, \end{aligned} \tag{4}$$

where C in $\mathcal{L}_{\text{GIoU}}$ denotes the smallest convex shape enclosing B and B^{gt}; $\boldsymbol{b}$ and $\boldsymbol{b}^{gt}$ in $\mathcal{L}_{\text{DIoU}}$ denote central points of B and B^{gt} with $\rho(\cdot)$ being the Euclidean distance and c being the diagonal length of the smallest enclosing box; and in $\mathcal{L}_{\text{CIoU}}$, $v = \frac{4}{\pi^2}(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h})^2$, $\beta = \frac{v}{(1-IoU)+v}$. They give us the family of power IoU losses for bbox regression with their original versions recovered at $\alpha = 1$. Note that the above α-IoU generalization can be easily extended to more complex loss functions that have multiple IoU or penalty terms (e.g., $\mathcal{L}_{\alpha\text{-CIoU}}$). Next, we will analyze properties of α-IoU losses when α takes different values.

3.3 Properties of α-IoU Losses

Here, we focus on the vanilla α-IoU formula $\mathcal{L}_{\alpha\text{-IoU}} = 1 - IoU^{\alpha}$ to analyze its properties, as different penalty terms may affect these properties in different manners. Figure 1 illustrates the correlation between IoU and $\mathcal{L}_{\alpha\text{-IoU}}$ (left) and the magnitude of its gradient w.r.t. IoU, i.e., $|\nabla_{\text{IoU}}\mathcal{L}_{\alpha\text{-IoU}}|$ (right). One key observation is that the IoU loss (i.e., $\alpha = 1$) has a linear correlation with IoU and the gradient is a constant, while $\mathcal{L}_{\alpha\text{-IoU}}$ re-weights objects adaptively (according to their IoU values) following different reweighting schemes with $0 < \alpha < 1$ versus $\alpha > 1$.

The power transformation in $\mathcal{L}_{\alpha\text{-IoU}}$ preserves key properties of $\mathcal{L}_{\text{IoU}}$ as a performance metric, including non-negativity, identity of indiscernibles, symmetry, and triangle inequality [28]. Furthermore,

we analyze following important properties of $\mathcal{L}_{\alpha\text{-IoU}}$ with detailed derivations deferred to Appendix A. We first let B_i and B_j be two predicted bboxes by two different models M_i and M_j respectively, and B_i and B_j correspond to the same ground truth B^{gt} with $IoU(B_i, B^{gt}) < IoU(B_j, B^{gt})$. Then we have the first property of $\mathcal{L}_{\alpha\text{-IoU}}$:

Property 1 (Order Preservingness)**.** *$\mathcal{L}_{\alpha\text{-}IoU}$ preserves orders of both IoU and $\mathcal{L}_{IoU}$: $IoU(B_i, B^{gt}) < IoU(B_j, B^{gt}) \iff \mathcal{L}_{IoU}(B_i, B^{gt}) > \mathcal{L}_{IoU}(B_j, B^{gt}) \iff \mathcal{L}_{\alpha\text{-}IoU}(B_i, B^{gt}) > \mathcal{L}_{\alpha\text{-}IoU}(B_j, B^{gt})$.*

The above property indicates that both $\mathcal{L}_{\alpha\text{-IoU}}$ and $\mathcal{L}_{\text{IoU}}$ are monotonically decreasing functions w.r.t. IoU. As $\mathcal{L}_{\alpha\text{-IoU}}$ preserves the order of $\mathcal{L}_{\text{IoU}}$ strictly, it is guaranteed that $\arg\min_B \mathcal{L}_{\alpha\text{-IoU}}(B, B^{gt})$ is identical to $\arg\max_B IoU(B, B^{gt})$ and $\arg\min_B \mathcal{L}_{\text{IoU}}(B, B^{gt})$. In other words, the optimal solution $\arg\max_B IoU(B, B^{gt})$ can be obtained by minimizing either $\mathcal{L}_{\alpha\text{-IoU}}$ or $\mathcal{L}_{\text{IoU}}$. Following this, the adaptive relative loss reweighting scheme of $\mathcal{L}_{\alpha\text{-IoU}}$ can be characterized by the second property:

Property 2 (Relative Loss Reweighting)**.** *Compared with $\mathcal{L}_{IoU}$, $\mathcal{L}_{\alpha\text{-}IoU}$ adaptively re-weights the relative loss of all objects by $w_{\mathcal{L}_r} = \mathcal{L}_{\alpha\text{-}IoU}/\mathcal{L}_{IoU} = 1 + (IoU - IoU^{\alpha})/(1 - IoU)$, with $w_{\mathcal{L}_r}(IoU = 0) = 1$, and $\lim_{IoU \to 1} w_{\mathcal{L}_r} = \alpha$.*

The second property indicates that $\mathcal{L}_{\alpha\text{-IoU}}$ will adaptively down-weight and up-weight the relative loss of all objects according to their IoUs when $0 < \alpha < 1$ and $\alpha > 1$, respectively. We further note that, when $\alpha > 1$, the reweighting factor $w_{\mathcal{L}_r}$ increases monotonically with the increase of IoU ($w_{\mathcal{L}_r}$ grows from 1 to α) while decreasing monotonically with the increase of IoU when $0 < \alpha < 1$ ($w_{\mathcal{L}_r}$ decays from 1 to α). We will empirically show that the up-weighting scheme of $\mathcal{L}_{\alpha\text{-IoU}}$ with $\alpha > 1$ can help the model focus more on high IoU objects to improve both the localization (i.e., predict more high IoU objects) and detection (i.e., more accurate at high APs) performance[2]. Similarly, we can obtain the third property of adaptive relative gradient reweighting owned by $\mathcal{L}_{\alpha\text{-IoU}}$ as follows:

Property 3 (Relative Gradient Reweighting)**.** *Compared with $\mathcal{L}_{IoU}$, $\mathcal{L}_{\alpha\text{-}IoU}$ adaptively re-weights the relative gradient of all objects by $w_{\nabla_r} = |\nabla_{IoU}\mathcal{L}_{\alpha\text{-}IoU}|/|\nabla_{IoU}\mathcal{L}_{IoU}| = \alpha IoU^{\alpha-1}$, with the turning point at $IoU = \alpha^{\frac{1}{1-\alpha}} \in (0, \frac{1}{e})$ when $0 < \alpha < 1$ and $IoU = \alpha^{\frac{1}{1-\alpha}} \in (\frac{1}{e}, 1)$ when $\alpha > 1$.*

When $\alpha > 1$, the above reweighting factor w_{∇_r} increases monotonically with the increase of IoU, while decreasing monotonically with the increase of IoU when $0 < \alpha < 1$. This relative gradient re-weighting scheme is also adaptive to IoU, with the turning point from up-weighting to down-weighting at $IoU = \alpha^{\frac{1}{1-\alpha}} \in (0, \frac{1}{e})$ when $0 < \alpha < 1$, and from down-weighting to up-weighting at $IoU = \alpha^{\frac{1}{1-\alpha}} \in (\frac{1}{e}, 1)$ when $\alpha > 1$. This relative gradient re-weighting scheme allows the model to learn objects with adaptive speeds (i.e., different gradients) according to their IoUs. Theoretically, when $\alpha = 2$, $|\nabla_{\text{IoU}}\mathcal{L}_{\alpha\text{-IoU}}| > |\nabla_{\text{IoU}}\mathcal{L}_{\text{IoU}}|$ for $IoU \in (0.5, 1]$, which accelerates the learning of all positive IoU objects at AP_{50}. However, we empirically show that α-IoU losses with $\alpha = 3$ perform more competitively than those with $\alpha = 2$ in most cases. It is probable that α-IoU losses with $\alpha = 3$ further up-weight the relative loss of objects with $IoU \in (0.5, 1]$, although α-IoU losses with $\alpha = 2$ also beat existing baselines (see Figure 3).

The above loss and gradient reweighting schemes can also be inferred from Figure 1, with detailed proofs in Appendix A. To summarize, $\mathcal{L}_{\alpha\text{-IoU}}$ trains better detectors than $\mathcal{L}_{\text{IoU}}$ for at least three reasons. First, the same optimal IoU can be achieved by $\mathcal{L}_{\alpha\text{-IoU}}$ as that by $\mathcal{L}_{\text{IoU}}$ (**Property 1**). Second, $\mathcal{L}_{\alpha\text{-IoU}}$ with $\alpha > 1$ focuses more on high IoU objects by up-weighting their relative loss (**Property 2**). Third, $\mathcal{L}_{\alpha\text{-IoU}}$ with $\alpha > 1$ helps detectors learn faster on high IoU objects (here $IoU \in (\alpha^{\frac{1}{1-\alpha}}, 1]$) through up-weighting their relative gradient (**Property 3**). We argue that such adaptive reweighting schemes provide more flexibility in achieving different levels of bbox regression accuracy (AP measured under different IoU thresholds). We will empirically show, in Figure 2c and 2d, that up-weighting the loss and gradient of high IoU objects can boost the training of the detector at a late stage. As a comparison, we will also show that α-IoU losses with $0 < \alpha < 1$ tend to degrade the detection performance in Section 4.4. Reducing the relative loss and gradient of high IoU objects ends up with more objects poorly localized.

[2]The relative loss reweighting scheme of $\mathcal{L}_{\alpha\text{-IoU}}$ is reminiscent of the reweighting scheme of the focal loss $FL(\hat{p}) = -(1-\hat{p})^{\gamma}\log(\hat{p}),\ \gamma > 0$, which was proposed to encourage the learning of low confidence foreground objects in the presence of a large number of high confidence backgrounds [19]. In contrast, $\mathcal{L}_{\alpha\text{-IoU}}$ with $\alpha > 1$ focuses more on high IoU objects as the final performance is comprehensively measured by $\text{mAP}_{50:95}$, with low IoU objects (here $IoU < 0.5$) suppressed at evaluation. Note that focal loss is usually designed for the classification branch of object detectors while our α-IoU is for the bbox regression branch.

Table 1: The performance of YOLOv5s, YOLOv5x and DETR models trained using different localization losses on PASCAL VOC and MS COCO benchmarks. Results are obtained on the test set of PASCAL VOC 2007 and the val set of MS COCO 2017. mAP denotes $\text{mAP}_{50:95}$; $\text{mAP}_{75:95}$ denotes the mean AP over $\text{AP}_{75}, \text{AP}_{80}, \cdots, \text{AP}_{95}$. "rela. improv." stands for relative improvement. $\alpha = 3$ is used for all α-IoU losses in all experiments.

Method	Loss	PASCAL VOC						MS COCO					
		AP_{50}	AP_{75}	AP_{85}	AP_{95}	**mAP**	$\textbf{mAP}_{75:95}$	AP_{50}	AP_{75}	AP_{85}	AP_{95}	**mAP**	$\textbf{mAP}_{75:95}$
YOLOv5s	$\mathcal{L}_{\text{IoU}}$	78.63	57.85	34.98	2.34	52.57	32.36	55.51	38.59	23.58	2.07	36.29	21.82
	$\mathcal{L}_{\alpha\text{-IoU}}$	78.49	58.61	38.07	3.63	53.48	34.37	55.25	39.69	25.85	3.35	37.01	23.66
	rela. improv.	**-0.18%**	**1.31%**	**8.83%**	**55.13%**	**1.73%**	**6.22%**	**-0.47%**	**2.85%**	**9.63%**	**61.84%**	**1.98%**	**8.43%**
	$\mathcal{L}_{\text{DIoU}}$	77.99	57.54	34.74	2.36	52.12	32.04	55.67	39.01	23.56	2.03	36.36	21.95
	$\mathcal{L}_{\alpha\text{-DIoU}}$	78.42	59.14	38.31	3.50	53.70	34.57	55.84	39.49	25.49	3.30	36.74	23.34
	rela. improv.	**0.55%**	**2.78%**	**10.28%**	**48.31%**	**3.03%**	**7.92%**	**0.31%**	**1.23%**	**8.19%**	**62.56%**	**1.05%**	**6.32%**
YOLOv5x	$\mathcal{L}_{\text{IoU}}$	85.24	70.08	53.08	10.88	63.95	46.78	67.36	52.15	38.22	9.31	48.42	34.42
	$\mathcal{L}_{\alpha\text{-IoU}}$	84.83	70.20	53.75	13.74	64.25	48.06	67.72	52.61	38.62	9.76	48.67	34.72
	rela. improv.	**-0.48%**	**0.17%**	**1.26%**	**26.29%**	**0.47%**	**2.73%**	**0.53%**	**0.88%**	**1.05%**	**4.83%**	**0.52%**	**0.87%**
	$\mathcal{L}_{\text{DIoU}}$	85.04	71.05	53.71	11.11	64.21	47.30	67.54	52.03	38.02	8.58	48.38	34.16
	$\mathcal{L}_{\alpha\text{-DIoU}}$	84.90	71.34	54.23	13.85	64.49	48.40	67.42	52.65	39.28	10.29	48.81	35.42
	rela. improv.	**-0.16%**	**0.41%**	**0.97%**	**24.66%**	**0.44%**	**2.32%**	**-0.18%**	**1.19%**	**3.31%**	**19.93%**	**0.89%**	**3.68%**
DETR	$\mathcal{L}_{\text{IoU}}$	76.50	53.85	29.54	1.62	49.78	28.82	59.38	41.67	26.13	3.52	39.23	24.37
	$\mathcal{L}_{\alpha\text{-IoU}}$	76.22	55.03	32.30	2.28	51.12	31.08	59.61	42.65	28.57	5.09	40.18	26.44
	rela. improv.	**-0.37%**	**2.19%**	**9.34%**	**40.74%**	**2.69%**	**7.84%**	**0.39%**	**2.35%**	**9.34%**	**44.60%**	**2.42%**	**8.49%**
	$\mathcal{L}_{\text{DIoU}}$	76.26	54.09	29.23	1.56	49.91	28.68	59.28	41.62	26.09	3.54	39.25	24.48
	$\mathcal{L}_{\alpha\text{-DIoU}}$	76.44	54.89	31.48	2.44	50.96	30.60	59.38	42.34	28.23	5.36	39.94	26.05
	rela. improv.	**0.24%**	**1.48%**	**7.70%**	**56.41%**	**2.10%**	**6.69%**	**0.17%**	**1.73%**	**8.20%**	**51.41%**	**1.76%**	**6.41%**

4 Experiments

4.1 Datasets and Training Setup

We conduct all experiments on two popular benchmarks, i.e., PASCAL VOC [9] and MS COCO [20]. On the PASCAL VOC benchmark, we train all models on the trainval set 2007+2012 (containing $16,551$ images from 20 categories) and evaluate them on the test set 2007 (containing $4,952$ images) [9]. On the MS COCO benchmark, we train all models on the training set 2017 (containing 118K images from 80 categories) and evaluate them on the val set 2017 (containing 5K images) [20]. We train all state-of-the-art models with the original implementation released by the authors. Specifically, we follow the original implementation's training protocol with default parameters and the number of training epochs with different losses [27, 28, 38, 4]. Implementation details of all models are given in Appendix B.1. All experiments are implemented using NVIDIA V100 GPUs.

4.2 Results and Analysis

We first validate the effectiveness of α-IoU losses in training both anchor-based and anchor-free models on above two datasets. We choose YOLOv5s (i.e., YOLOv5 small) and YOLOv5x (i.e., YOLOv5 extra large) as one-stage anchor-based models, and DETR (ResNet-50) as an anchor-free model. Both α-IoU losses (i.e., $\mathcal{L}_{\alpha\text{-IoU}}$ and $\mathcal{L}_{\alpha\text{-DIoU}}$) are generalized from existing baselines following equation (4). From Table 1, we can observe that α-IoU losses surpass existing losses consistently across multiple models and datasets in terms of both mAP and $\text{mAP}_{75:95}$, especially at the high bbox regression accuracy $\text{mAP}_{75:95}$. The superiority of α-IoU losses is more pronounced at high accuracy levels, which might reach 60% relative improvement at AP_{95}. Interestingly, α-IoU losses tend to help more of light models (e.g., YOLOv5s with 7.3M parameters and 17 GFLOPs) than heavy models (e.g., YOLOv5x with 87.7M parameters and 218.8 GFLOPs). This indicates that α-IoU losses hold more advantage while training light models in computing-resource-limited scenarios, such as mobile devices, autonomous vehicles, and robots. Significant improvements on both PASCAL VOC and MS COCO demonstrate the stability of α-IoU losses across different datasets. In addition, we also verify its robustness to extremely small training sets in Appendix B.2.2, where α-IoU losses beat existing losses at varying scales, i.e., from 4K (25% trainval set of PASCAL VOC 2007+2012) to 118K (the entire training set of MS COCO 2017) samples. It is possible that α-IoU losses may not perform well if measured by a single low AP metric. For example, there may be less than 0.5% performance drop at AP_{50}, however, this is compensated by the significant boost at high APs. With some examples in

Table 2: The performance of Faster R-CNN (ResNet-50-FPN) with $1\times$ schedule and single scale training on MS COCO using different localization losses. Results are obtained on the val set of MS COCO 2017. mAP denotes $\text{mAP}_{50:95}$; $\text{mAP}_{75:95}$ denotes the mean AP over $\text{AP}_{75}, \text{AP}_{80}, \cdots, \text{AP}_{95}$. AP_s, AP_m, and AP_l denote the AP for small, medium, and large objects, respectively. † marks results from the MMDetection toolbox [6], while * marks results in original papers. "–" represents no reported results in papers. "rela. improv." stands for relative improvement. $\alpha = 3$ is used for all α-IoU losses in all experiments. The top two best results in every column are **boldfaced**.

Loss	AP_{50}	AP_{75}	AP_{80}	AP_{85}	AP_{90}	AP_{95}	**mAP**	**mAP**$_{75:95}$	AP_s	AP_m	AP_l
$^\dagger \ell_1$	58.13	40.45	33.56	23.39	11.09	1.24	37.37	21.95	21.20	40.96	48.13
$^\dagger \mathcal{L}_{\text{IoU}}$	58.12	41.23	34.03	24.43	12.42	1.61	37.88	22.74	21.61	41.63	49.11
$^\dagger \mathcal{L}_{\text{GIoU}}$	58.18	41.00	33.52	24.13	11.97	1.51	37.62	22.43	21.49	41.07	48.90
$^\dagger \mathcal{L}_{\text{BIoU}}$	58.05	40.57	33.54	23.85	11.10	1.19	37.43	22.05	21.57	41.00	48.17
$^* \mathcal{L}_{\text{IoU}}$	–	40.79	–	–	–	–	37.93	–	21.58	40.82	50.14
$^* \mathcal{L}_{\text{GIoU}}$	–	41.11	–	–	–	–	38.02	–	21.45	41.06	50.21
$^* \mathcal{L}_{\text{DIoU}}$	–	41.11	–	–	–	–	38.09	–	21.66	41.18	50.32
$^* \mathcal{L}_{\text{CIoU}}$	–	41.96	–	–	–	–	38.65	–	21.32	41.83	**51.51**
$^* \mathcal{L}_{\text{Focal-EIoU}}$	**59.10**	**42.40**	–	–	–	–	38.90	–	21.20	41.10	50.20
*Autoloss	58.60	41.80	–	–	–	–	38.50	–	22.00	**42.20**	50.20
$\mathcal{L}_{\alpha\text{-IoU}}$	58.81	41.94	34.81	**25.36**	**13.27**	1.81	38.96	23.44	**22.14**	42.11	50.36
$\mathcal{L}_{\alpha\text{-GIoU}}$	59.01	42.00	**35.13**	25.14	13.09	**2.03**	39.18	**23.46**	22.05	**42.19**	50.08
$\mathcal{L}_{\alpha\text{-DIoU}}$	**59.27**	**42.18**	**35.25**	**25.47**	**13.32**	1.95	**39.43**	**23.65**	**22.10**	42.10	**50.43**
$\mathcal{L}_{\alpha\text{-CIoU}}$	59.09	41.92	35.01	25.08	13.04	**1.98**	**39.25**	23.41	21.94	41.88	50.01

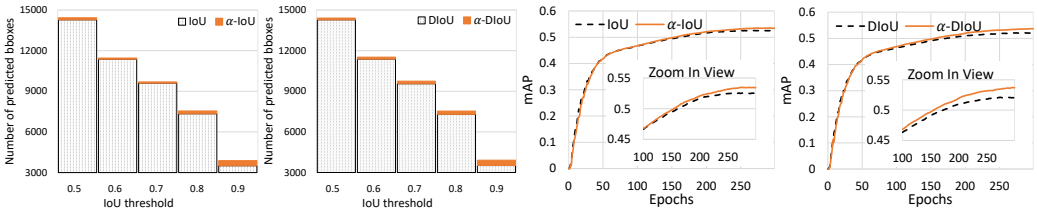

Figure 2: The left two plots are the IoU distributions between predicted bboxes and their ground truth after NMS with an IoU threshold 0.5. The right two plots are the validation mAPs ($\text{mAP}_{50:95}$) across 300 training epochs. All results are obtained from YOLOv5s trained on PASCAL VOC.

Figure 4 and more in Appendix B.3, we show that α-IoU losses are able to localize objects more accurately than baselines, thus help detect more true positive and fewer false positive objects.

We analyze the bbox regression accuracy by showing the IoU distributions between predicted bboxes and their ground truth from the detector trained by different losses. After NMS with the IoU threshold being 0.5, we further visualize the number of positive predicted bboxes under different IoU thresholds from 0.5 to 0.9 in the left two plots in Figure 2, showing that α-IoU losses detect more positive objects than baseline losses across all IoU thresholds. It is worth mentioning that α-IoU losses detect approximately 1% more positive objects than baselines when $IoU \geq 0.5$, and 11% more high IoU objects when $IoU \geq 0.9$. α-IoU is extremely advantageous in pushing low IoU objects to high IoU objects by up-weighting their relative loss, thus outperforming baseline losses significantly at the high accuracy level, and contributing to further improvement of the final detection performance (Table 1). Moreover, the right two plots in Figure 2 show that α-IoU losses are able to boost the late training stage (e.g., after 200 epochs) through up-weighting the loss and gradient of high IoU objects, while almost having no negative impact on the early training stage (e.g., first 100 epochs).

We also conduct experiments to compare extensive IoU-based losses using the popular two-stage anchor-based model, i.e., Faster R-CNN (ResNet-50-FPN). In Table 2, results at the top are reproduced from the MMDetection toolbox [6] while those in the middle are reported results from original papers [38, 36, 21]. Results at the bottom are obtained by replacing existing losses with α-IoU losses for Faster R-CNN (ResNet-50-FPN) on MMDetection (i.e., improve based on top results). Experimental results on MS COCO show that α-IoU losses are quite competitive compared with existing baselines in terms of both mAP and $\text{mAP}_{75:95}$. Note that the Autoloss searches both the classification loss and the localization loss, thus taking a huge amount of searching time [21]. In contrast, α-IoU losses only change the localization loss and win the Autoloss without additional computational overhead.

Table 3: The performance of YOLOv5s trained using different localization losses on simulated noisy trainval sets of PASCAL VOC 2007+2012 under noise rates $\eta = 0.1, 0.2$, and 0.3. Results are obtained on the clean test set of PASCAL VOC 2007. mAP denotes $\text{mAP}_{50:95}$; $\text{mAP}_{75:95}$ denotes the mean AP over $\text{AP}_{75}, \text{AP}_{80}, \cdots, \text{AP}_{95}$. "rela. improv." stands for relative improvement. $\alpha = 3$ is used for all α-IoU losses in all experiments.

Noise	Loss	AP_{50}	AP_{55}	AP_{60}	AP_{65}	AP_{70}	AP_{75}	AP_{80}	AP_{85}	AP_{90}	AP_{95}	**mAP**	**mAP**$_{75:95}$
0.1	$\mathcal{L}_{\text{IoU}}$	74.48	71.57	68.08	63.29	56.55	47.12	33.06	17.53	4.16	0.26	43.61	20.43
	$\mathcal{L}_{\alpha\text{-IoU}}$	74.67	71.94	68.73	64.27	57.75	48.50	36.88	21.25	6.30	0.28	45.06	22.64
	rela. improv.	**0.26%**	**0.52%**	**0.95%**	**1.55%**	**2.12%**	**2.93%**	**11.55%**	**21.22%**	**51.44%**	**7.69%**	**3.32%**	**10.85%**
	$\mathcal{L}_{\text{DIoU}}$	74.09	71.46	67.88	63.09	56.18	46.71	32.67	17.50	4.43	0.23	43.42	20.31
	$\mathcal{L}_{\alpha\text{-DIoU}}$	74.38	71.95	68.10	63.52	57.18	48.47	35.90	20.89	6.37	0.33	44.71	22.39
	rela. improv.	**0.39%**	**0.69%**	**0.32%**	**0.68%**	**1.78%**	**3.77%**	**9.89%**	**19.37%**	**43.79%**	**43.48%**	**2.97%**	**10.26%**
0.2	$\mathcal{L}_{\text{IoU}}$	67.82	63.93	58.22	50.11	39.31	26.33	13.51	4.55	0.66	0.05	32.45	9.02
	$\mathcal{L}_{\alpha\text{-IoU}}$	68.20	64.21	58.77	51.59	40.66	29.20	16.11	6.06	1.31	0.10	33.62	10.56
	rela. improv.	**0.56%**	**0.44%**	**0.94%**	**2.95%**	**3.43%**	**10.90%**	**19.25%**	**33.19%**	**98.48%**	**100%**	**3.61%**	**17.03%**
	$\mathcal{L}_{\text{DIoU}}$	67.39	62.94	57.29	49.25	39.40	27.13	13.78	4.52	0.68	0.02	32.24	9.23
	$\mathcal{L}_{\alpha\text{-DIoU}}$	68.26	64.49	59.59	51.99	41.19	29.12	15.77	5.84	1.25	0.21	33.77	10.44
	rela. improv.	**1.29%**	**2.46%**	**4.01%**	**5.56%**	**4.54%**	**7.34%**	**14.44%**	**29.20%**	**83.82%**	**950%**	**4.75%**	**13.14%**
0.3	$\mathcal{L}_{\text{IoU}}$	56.54	49.69	40.67	30.80	19.99	11.13	4.81	1.43	0.31	0.04	21.54	3.54
	$\mathcal{L}_{\alpha\text{-IoU}}$	58.59	51.58	43.23	32.93	22.27	12.52	5.91	2.16	0.73	0.12	23.00	4.29
	rela. improv.	**3.63%**	**3.80%**	**6.29%**	**6.92%**	**11.41%**	**12.49%**	**22.87%**	**51.05%**	**135%**	**200%**	**6.78%**	**20.99%**
	$\mathcal{L}_{\text{DIoU}}$	56.84	49.82	41.50	32.06	20.80	11.22	4.84	1.51	0.46	0.07	21.91	3.62
	$\mathcal{L}_{\alpha\text{-DIoU}}$	58.45	51.94	43.9	33.78	22.57	12.89	6.34	2.42	0.65	0.16	23.31	4.49
	rela. improv.	**2.83%**	**4.26%**	**5.78%**	**5.36%**	**8.51%**	**14.88%**	**30.99%**	**60.26%**	**41.30%**	**129%**	**6.39%**	**24.09%**

4.3 Robustness to Noisy Bounding Boxes

It happens quite often that people annotate inaccurate bboxes in images/videos as the ground truth, even with computer-assisted annotation tools. However, there is little work on the robustness of localization losses to noisy bboxes, even though a number of methods have been proposed for the robust learning with noisy labels, anchors, and bboxes [10, 13, 17, 18]. Here, we fill this gap by conducting a set of experiments to evaluate the robustness of different localization losses to noisy bboxes, and show that α-IoU losses are more robust than other losses. Considering that open datasets like PASCAL VOC and MS COCO are carefully annotated, we synthesize a set of common noisy bboxes by perturbing normalized bboxes in the entire training set. The perturbations follow a uniform noise distribution in $[-\eta w, \eta w]$ at horizontal coordinates (x and w) and $[-\eta h, \eta h]$ at vertical coordinates (y and h), where η is the noise rate [18]. We then constrain all noisy bboxes within images by following boundary conditions:

$$0 < w < 1, \quad 0 < h < 1, \quad \frac{1}{2}w \leq x \leq 1 - \frac{1}{2}w, \quad \frac{1}{2}h \leq y \leq 1 - \frac{1}{2}h. \tag{5}$$

We set $\eta = 0.1, 0.2, 0.3$ in our experiments, with the average IoU between noisy bboxes and their corresponding clean ones dropping to 0.833, 0.710, and 0.613, respectively. Examples of synthesized noisy bboxes are given in Appendix B.4. As shown in Table 3, α-IoU improves baseline losses (i.e., $\mathcal{L}_{\text{IoU}}$ and $\mathcal{L}_{\text{DIoU}}$) considerably in noisy scenarios. We gain increasing relative improvements from AP_{50} to AP_{95}, which accumulate to a more significant improvement in $\text{mAP}_{75:95}$. Note that α-IoU losses also outperform baselines at AP_{50} across all noisy scenarios, which is not always the case when bboxes are clean (Table 1). Furthermore, α-IoU losses are noticeably more robust in more severely noisy scenarios. For example, the relative improvement of $\mathcal{L}_{\alpha\text{-DIoU}}$ over $\mathcal{L}_{\text{DIoU}}$ increases from $2.97\%/10.26\%$ to $6.39\%/24.09\%$ according to mAP/$\text{mAP}_{75:95}$ when the noise rate η rises from 0.1 to 0.3. These results confirm the advantage of α-IoU losses in noisy bbox scenarios.

4.4 Sensitivity to power parameter α

Here, we evaluate the performance of α-IoU with varying α values via a set of experiments with $\mathcal{L}_{\alpha\text{-IoU}}$ and $\mathcal{L}_{\alpha\text{-DIoU}}$ from equation (4) and $\alpha \in [0.5, 5]$, as shown in Figure 3. The results are obtained using YOLOv5s on PASCAL VOC in both clean and various noisy bbox scenarios. It can be seen that α-IoU losses with $\alpha \in [2, 4]$ perform competitively well across all scenarios, with $\alpha = 3$ performing the best in most cases. In addition, we also observe that α-IoU losses with $\alpha < 1$ perform consistently worse than baselines (i.e., α-IoU with $\alpha = 1$), validating the necessity of up-weighting the loss and gradient of high IoU objects, rather than down-weighting. We also test an extreme case with $\alpha = 10$, in which the performance drops by $5.61\%/10.92\%/23.88\%/31.82\%$ when $\eta = 0/0.1/0.2/0.3$ on

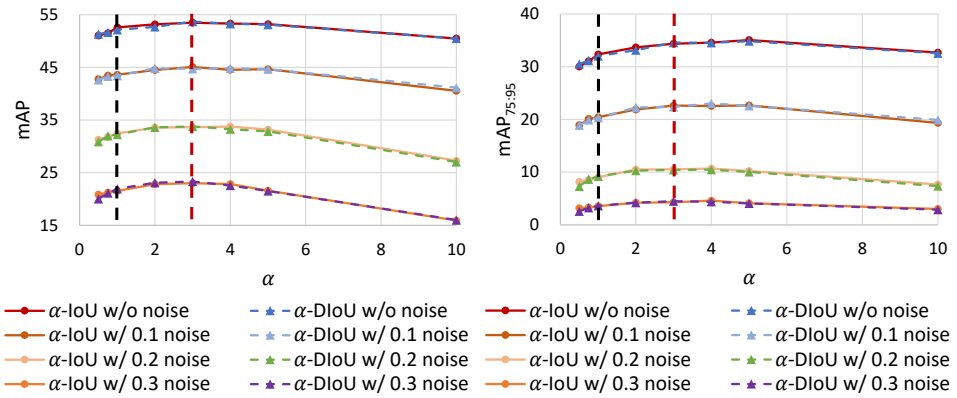

Figure 3: Comparison of the performance of YOLOv5s using α-IoU with varying α values. Results are obtained on the clean test set of PASCAL VOC 2007. Black dashed lines denote baselines (i.e., the family of α-IoU with $\alpha = 1$) while red dashed lines denote the family of α-IoU with $\alpha = 3$.

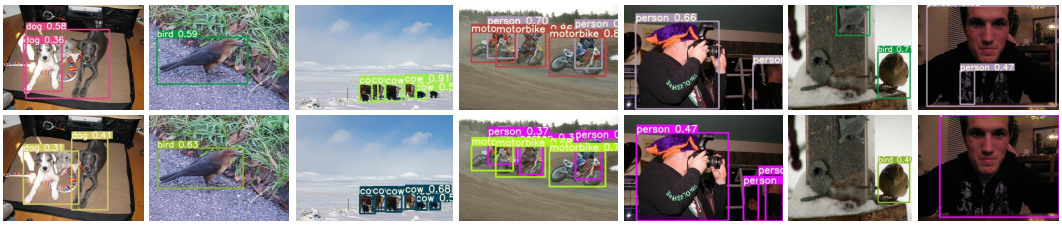

Figure 4: Example results on the test set of PASCAL VOC 2007 using YOLOv5s trained by $\mathcal{L}_{\text{IoU}}$ (top row) and $\mathcal{L}_{\alpha\text{-IoU}}$ with $\alpha = 3$ (bottom row). $\mathcal{L}_{\alpha\text{-IoU}}$ performs better than $\mathcal{L}_{\text{IoU}}$ because it can localize objects more accurately (image 1 and 2), thus can detect more true positive objects (image 3 to 5) and fewer false positive objects (image 6 and 7).

average compared with $\alpha = 3$, and is even worse than baselines. This indicates that a proper choice of α is crucial, and the optimal α is 3 at least for tested models and datasets.

5 Conclusions

In this paper, we propose α-IoU, a unified formula that generalizes existing IoU-based losses to a new family of power IoU losses. By modulating a single power parameter α, α-IoU can help achieve different levels of bbox regression accuracy. We analyze the order preservingness and the relative loss/gradient reweighting properties of α-IoU to show that α-IoU can improve bbox regression accuracy through up-weighting the relative loss and gradient of high IoU objects. Experiments with multiple models and benchmark datasets demonstrate that α-IoU losses can significantly outperform existing IoU losses, especially at high accuracy levels. α-IoU is potential to be widely applied in practical scenarios as, 1) it significantly benefit light models; 2) it is extremely powerful on small datasets; and more importantly, 3) it is more robust to noisy bboxes. For future work, we will investigate if the α-IoU type generalization strategy can improve other metric-derived loss functions, e.g., Dice, Hausdorff distance, and Chamfer distance losses.

Societal Impacts

Our work can help train high-performance object detection and localization models for impactful applications such as self-driving, face recognition and video surveillance. While not our initial intention, these models might be manipulated by adversaries or unauthorized users for adversarial or illegal purposes, which can potentially compromise the safety and privacy of individuals. We believe strict regulations and laws should be established to prevent such exploitation.

References

[1] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[2] G. E. Box and D. R. Cox. An analysis of transformations. *Journal of the Royal Statistical Society: Series B (Methodological)*, 26(2):211–243, 1964.

[3] Z. Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, pages 6154–6162, 2018.

[4] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer, 2020.

[5] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang, et al. Hybrid task cascade for instance segmentation. In *CVPR*, pages 4974–4983, 2019.

[6] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[7] Z. Dong, G. Li, Y. Liao, F. Wang, P. Ren, and C. Qian. Centripetalnet: Pursuing high-quality keypoint pairs for object detection. In *CVPR*, pages 10519–10528, 2020.

[8] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian. Centernet: Keypoint triplets for object detection. In *ICCV*, pages 6569–6578, 2019.

[9] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010.

[10] J. Gao, J. Wang, S. Dai, L.-J. Li, and R. Nevatia. Note-rcnn: Noise tolerant ensemble rcnn for semi-supervised object detection. In *ICCV*, pages 9508–9517, 2019.

[11] R. Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015.

[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017.

[13] M. Khodabandeh, A. Vahdat, M. Ranjbar, and W. G. Macready. A robust learning approach to domain adaptive object detection. In *ICCV*, pages 480–490, 2019.

[14] H. Law and J. Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, pages 734–750, 2018.

[15] H. Li, T. Fu, J. Dai, H. Li, G. Huang, and X. Zhu. Autoloss-zero: Searching loss functions from scratch for generic tasks. *arXiv preprint arXiv:2103.14026*, 2021.

[16] H. Li, C. Tao, X. Zhu, X. Wang, G. Huang, and J. Dai. Auto seg-loss: Searching metric surrogates for semantic segmentation. *arXiv preprint arXiv:2010.07930*, 2020.

[17] H. Li, Z. Wu, C. Zhu, C. Xiong, R. Socher, and L. S. Davis. Learning from noisy anchors for one-stage object detection. In *CVPR*, pages 10588–10597, 2020.

[18] J. Li, C. Xiong, R. Socher, and S. Hoi. Towards noise-resistant object detection with noisy annotations. *arXiv preprint arXiv:2003.01285*, 2020.

[19] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017.

[20] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014.

[21] P. Liu, G. Zhang, B. Wang, H. Xu, X. Liang, Y. Jiang, and Z. Li. Loss function discovery for object detection via convergence-simulation driven search. *arXiv preprint arXiv:2102.04700*, 2021.

[22] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37. Springer, 2016.

[23] X. Ma, H. Huang, Y. Wang, S. Romano, S. Erfani, and J. Bailey. Normalized loss functions for deep learning with noisy labels. In *ICML*, pages 6543–6553. PMLR, 2020.

[24] M. A. Rahman and Y. Wang. Optimizing intersection-over-union in deep neural networks for image segmentation. In *ISVC*, pages 234–244. Springer, 2016.

[25] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, pages 7263–7271, 2017.

[26] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[27] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015.

[28] 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 *CVPR*, pages 658–666, 2019.

[29] G. Song, Y. Liu, and X. Wang. Revisiting the sibling head in object detector. In *CVPR*, pages 11563–11572, 2020.

[30] Z. Tian, C. Shen, H. Chen, and T. He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, pages 9627–9636, 2019.

[31] L. Tychsen-Smith and L. Petersson. Improving object localization with fitness nms and bounded iou loss. In *CVPR*, pages 6877–6885, 2018.

[32] K. Wang and L. Zhang. Single-shot two-pronged detector with rectified iou loss. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1311–1319, 2020.

[33] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, and J. Bailey. Symmetric cross entropy for robust learning with noisy labels. In *ICCV*, pages 322–330, 2019.

[34] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang. Unitbox: An advanced object detection network. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 516–520, 2016.

[35] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *CVPR*, pages 9759–9768, 2020.

[36] Y.-F. Zhang, W. Ren, Z. Zhang, Z. Jia, L. Wang, and T. Tan. Focal and efficient iou loss for accurate bounding box regression. *arXiv preprint arXiv:2101.08158*, 2021.

[37] Z. Zhang and M. R. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *arXiv preprint arXiv:1805.07836*, 2018.

[38] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren. Distance-iou loss: Faster and better learning for bounding box regression. In *AAAI*, pages 12993–13000, 2020.

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

[40] X. Zhou, J. Zhuo, and P. Krahenbuhl. Bottom-up object detection by grouping extreme and center points. In *CVPR*, pages 850–859, 2019.

[41] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] In Section 4.4.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] In Section 5.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] In the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] In Appendix B.1.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] In Section 4.1.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] In the supplemental material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] From github complying with the license.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] We use two popular object detection datasets.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]