单阶段目标检测

密集物体检测技术的演进

- 院系：人工智能学院
- 专业：工科试验班
- 答辩人：黄子豪

目录

1 研究背景

2 Focal Loss

3 FCOS

4 GFL-V1

5 GFL-V2

6 青稞

7 技术演进
个人思考

1

研究背景

密集物体检测的背景与核心挑战

1 研究背景 RESEARCH BACKGROUNDS

	单阶段目标检测	双阶段目标检测
算法逻辑	产生候选区域 => 位置精修 => 进行候选区域分类	不产生候选区域 => 直接产生物体的类别概率和位置坐标值
准确性	较低(COCO mAP ~30%)	较高(COCO mAP ~40%)
速度	较慢(fps 10~100+)	较快(fps 3~5)
计算资源的消耗	较少(YOLOv5s: 参数量7MB)	较高(Faster-RCNN: 参数量40MB)
常见算法	YOLO、G-CNN、SSD、RON	Fast-RCNN、FPN

1 研究背景

RESEARCH BACKGROUNDS

	单阶段目标检测	双阶段目标检测
算法逻辑	产生候选区域 => 位置精修 => 进行候选区域分类	不产生候选区域 => 直接产生物体的类别概率和位置坐标值
准确性	较低(COCO mAP ~30%)	较高(COCO mAP ~40%)
速度	较慢(fps 10~100+)	较快(fps 3~5)
计算资源的消耗	较少(YOLOv5s: 参数量7MB)	较高(Faster-RCNN: 参数量40MB)
常见算法	YOLO、G-CNN、SSD、RON	Fast-RCNN、FPN

是什么限制了单阶段目标检测的准确性呢？又该从哪些方向进行突破呢？

1 研究背景 RESEARCH BACKGROUNDS

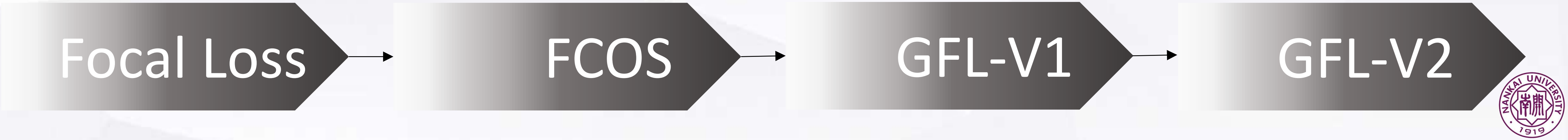
面临的核心挑战：

类别不平衡： 大量**易分类的背景样本**主导训练。

锚点框的束缚： 超参数敏感、设计复杂、对不规则物体适应性差。

定位质量与分类的不一致性： 训练与推理阶段使用方式不统一，**质量预测**不纯粹。

边界框表示的局限性： 传统回归难以表达**定位的不确定性。**



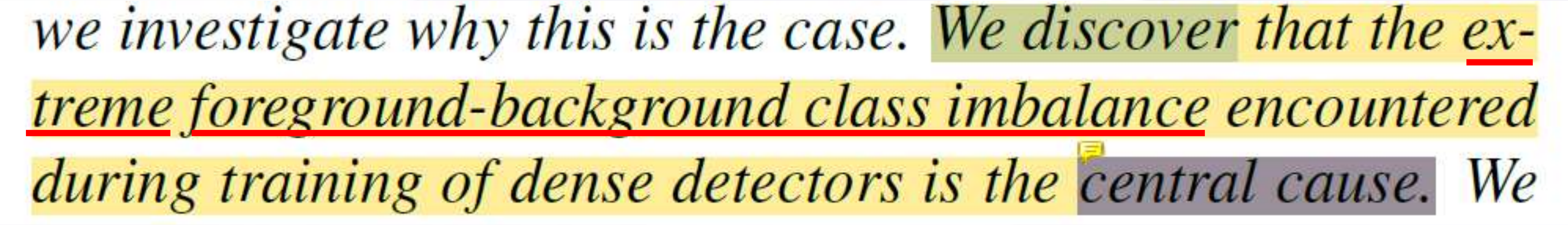
2

Focal Loss for Dense Object Detector

2017-ICCV

2 Focal Loss for Dense Object Detector

核心问题识别



we investigate why this is the case. We discover that the extreme foreground-background class imbalance encountered during training of dense detectors is the central cause. We

作者给出的解释是由于前正负样本不均衡的问题（简单-难分样本不均衡）

01

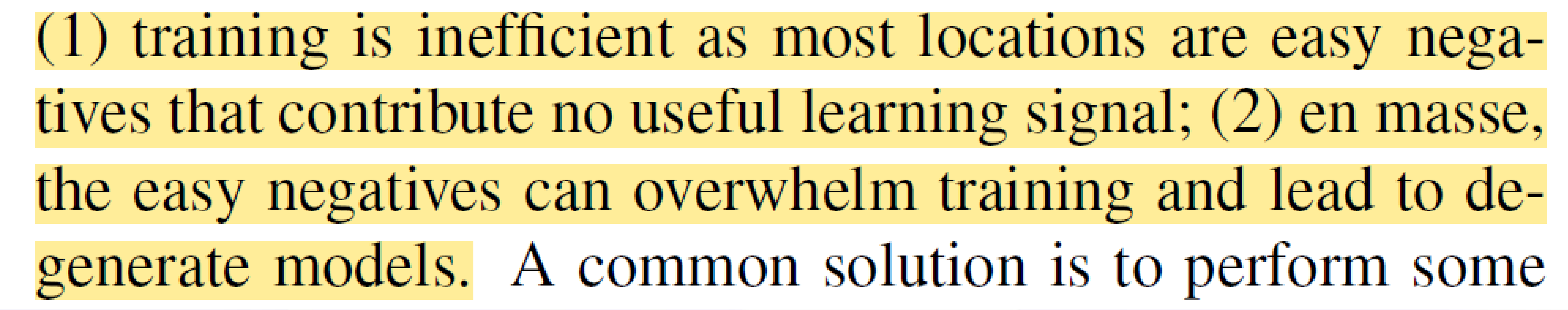
二分类：作者认为 foreground = negative， background = positive

02

样本不平均是**非常极端 extreme** 的。

2 Focal Loss for Dense Object Detector

样本类别不均衡会带来什么问题?



(1) training is inefficient as most locations are easy negatives that contribute no useful learning signal; (2) en masse, the easy negatives can overwhelm training and lead to degenerate models. A common solution is to perform some

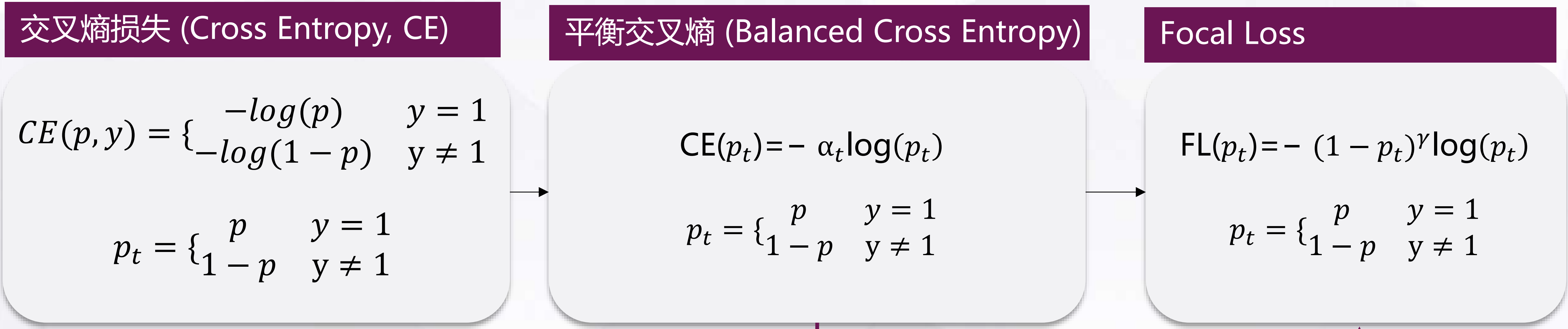
01 训练效率低下

大部分候选位置是**易于分类的负样本（背景）**，它们对学习信号的贡献很小。

02 模型退化

大量的易分类负样本在损失计算中占据主导地位，**淹没了少数正样本（前景）的信号**，可能导致模型学习效果不佳，甚至产生退化的模型。

2 Focal Loss for Dense Object Detector



交叉熵损失 (Cross Entropy, CE)

$$CE(p,y) = \begin{cases} -log(p) & y = 1 \\ -log(1-p) & y \neq 1 \end{cases}$$

$$p_t = \begin{cases} p & y = 1 \\ 1-p & y \neq 1 \end{cases}$$

平衡交叉熵 (Balanced Cross Entropy)

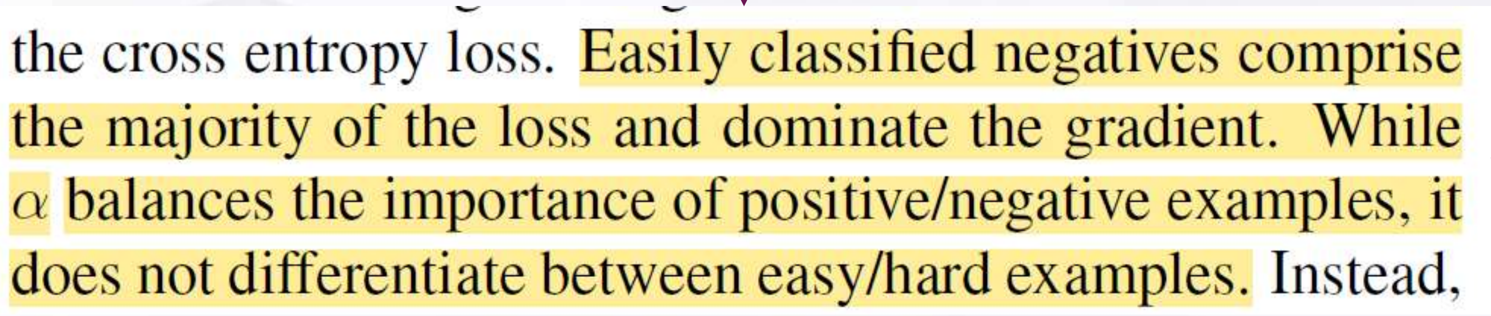
$$\text{CE}(p_t) = -\ \alpha_t \log(p_t)$$

$$p_t = \begin{cases} p & y = 1 \\ 1-p & y \neq 1 \end{cases}$$

Focal Loss

$$\text{FL}(p_t) = -\ (1-p_t)^{\gamma}\log(p_t)$$

$$p_t = \begin{cases} p & y = 1 \\ 1-p & y \neq 1 \end{cases}$$

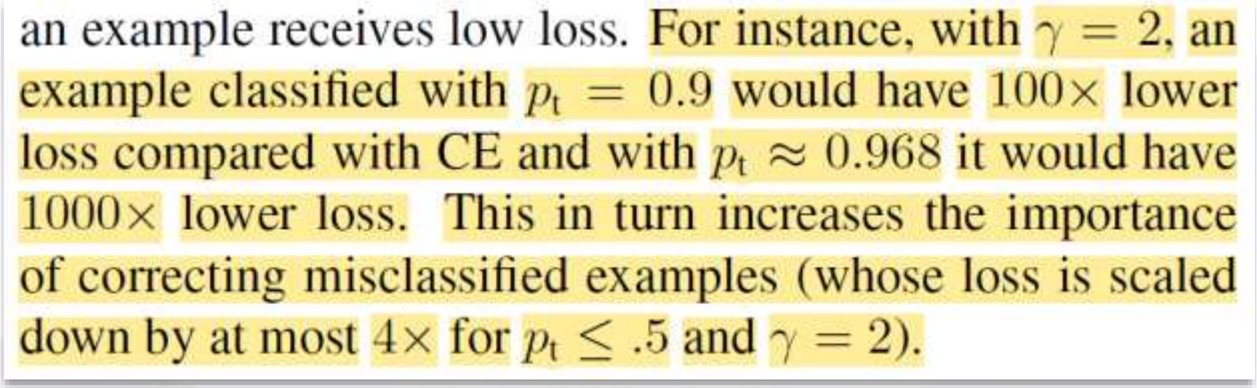


the cross entropy loss. Easily classified negatives comprise the majority of the loss and dominate the gradient. While α balances the importance of positive/negative examples, it does not differentiate between easy/hard examples. Instead,

2 Focal Loss for Dense Object Detector

$$\mathrm{FL}(\boxed{p_t}) = -\boxed{(1-p_t)}^{\boxed{\gamma}}\log(\boxed{p_t})$$

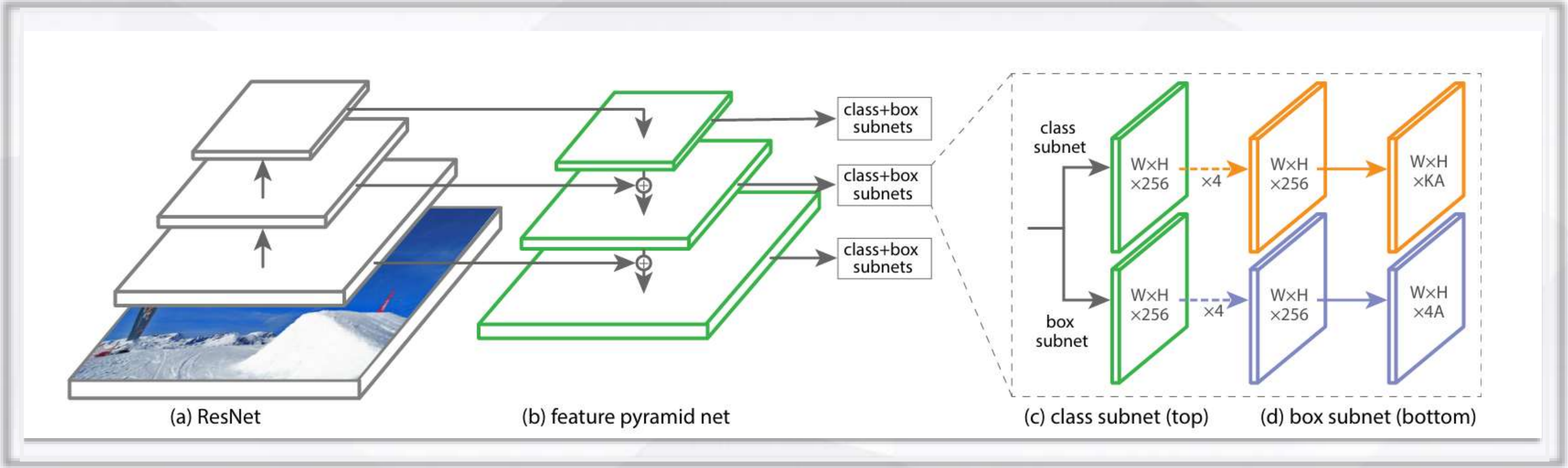
$(1-p_t)^{\gamma}$	调制因子 (modulating factor)
γ≥0	可调的聚焦参数 (focusing parameter)
pt	置信度



an example receives low loss. For instance, with $\gamma = 2$, an example classified with $p_t = 0.9$ would have 100× lower loss compared with CE and with $p_t \approx 0.968$ it would have 1000× lower loss. This in turn increases the importance of correcting misclassified examples (whose loss is scaled down by at most 4× for $p_t \leq .5$ and $\gamma = 2$).

p→1，正样本	Pt=p	1–pt→0	(1–pt)γ→0	log(p_t) →0	FL(p_t)→0
p→0，正样本	Pt=p	1–pt→1	(1–pt)γ→1	log(p_t) →-∞	FL(p_t)→+∞
p→1，负样本	Pt=1-p	1–pt→0	(1–pt)γ→0	log(p_t) →-∞	FL(p_t)→+∞
p→0，负样本	Pt=1-p	1–pt→1	(1–pt)γ→1	log(p_t) →0	FL(p_t)→0

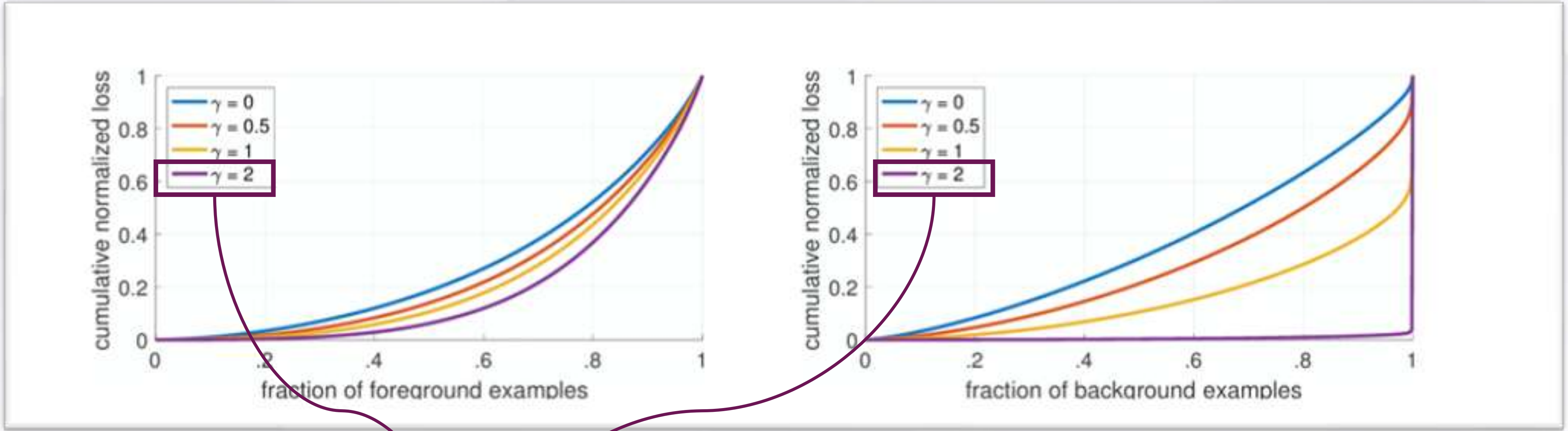
2 Focal Loss for Dense Object Detector

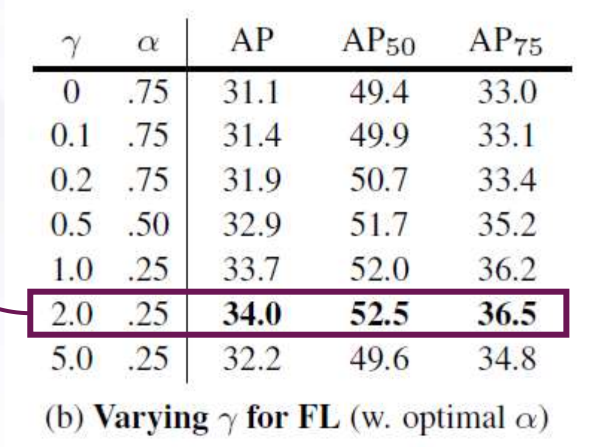


- **骨干网络**：ResNet + FPN (P3-P7)，提取多尺度特征。
- **锚点**：多尺度、多长宽比锚点 (每层9个)，IoU > 0.5为正。
- **分类子网络**：共享参数的FCN，预测类别 (KA个输出，Sigmoid)。
- **边界框回归子网络**：与分类并行，回归偏移量 (4A个输出)，类别无关。
- **推断**：前向传播 → 高分候选 → NMS。

2 Focal Loss for Dense Object Detector

探究了γ调节对模型性能影响





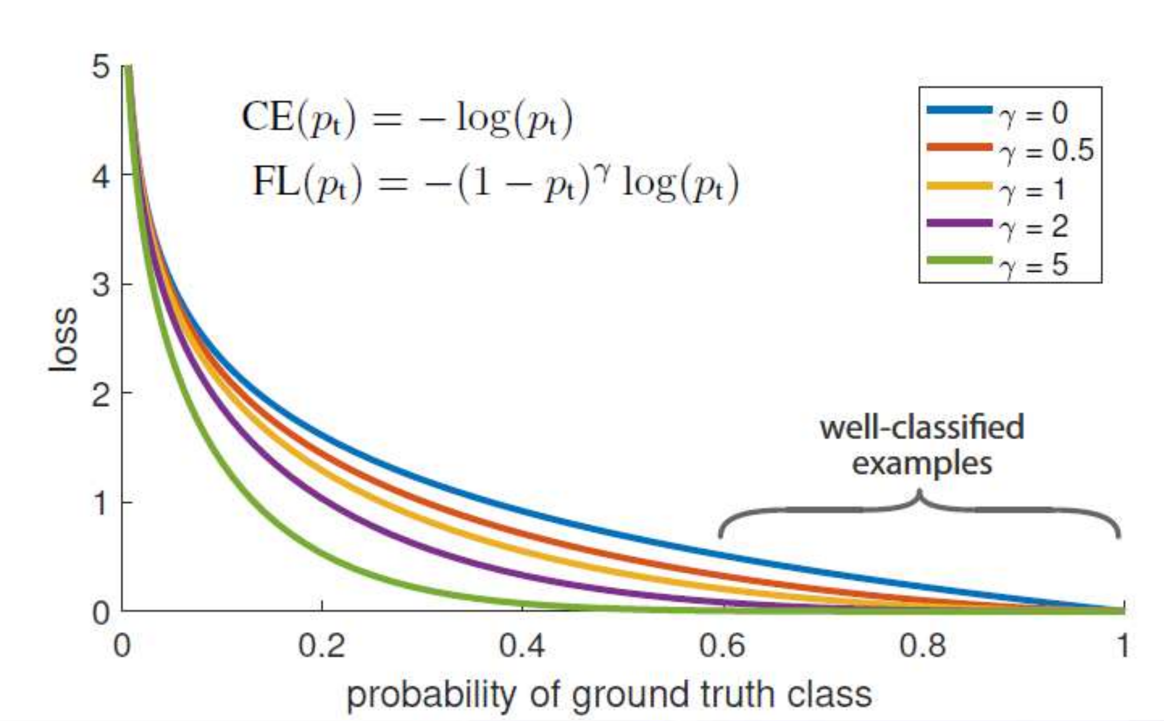
γ	α	AP	AP_{50}	AP_{75}
0	.75	31.1	49.4	33.0
0.1	.75	31.4	49.9	33.1
0.2	.75	31.9	50.7	33.4
0.5	.50	32.9	51.7	35.2
1.0	.25	33.7	52.0	36.2
2.0	.25	**34.0**	**52.5**	**36.5**
5.0	.25	32.2	49.6	34.8

(b) **Varying** γ **for FL** (w. optimal α)

γ=2时效果最好，γ过小起不到调节作用，过大则会对难易样本都过度抑制

2 Focal Loss for Dense Object Detector

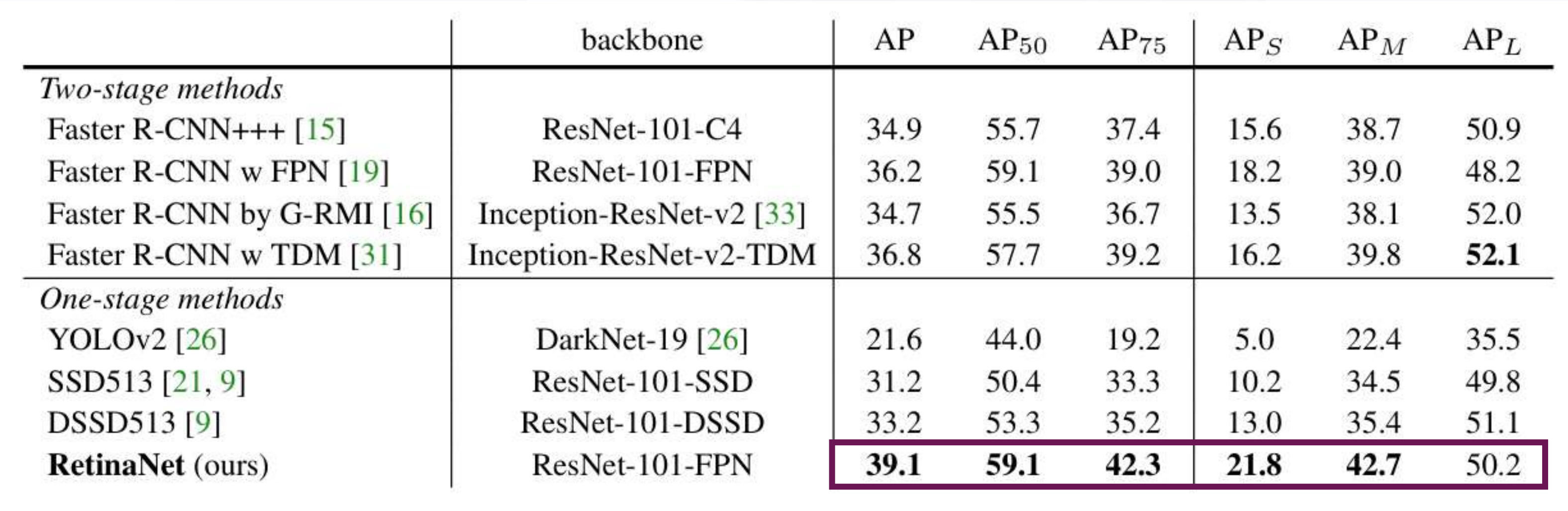
是否能区分易分/难分样本



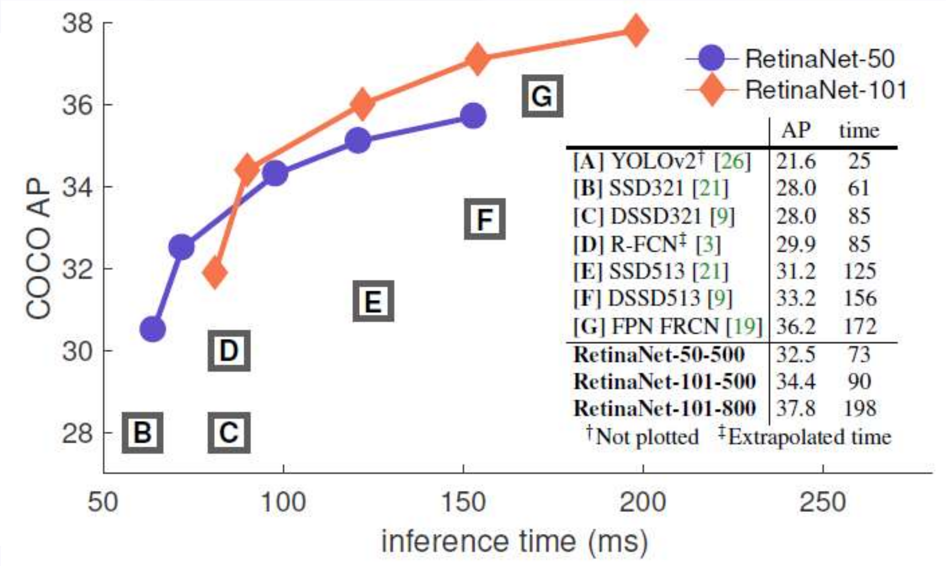
γ=2时效果最好，同时，更说明了添加了聚焦参数能让模型更准确的识别训练

2 Focal Loss for Dense Object Detector

优势进步所在



	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage methods							
Faster R-CNN+++ [15]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [19]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [16]	Inception-ResNet-v2 [33]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [31]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	**52.1**
One-stage methods							
YOLOv2 [26]	DarkNet-19 [26]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [21, 9]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [9]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet (ours)	ResNet-101-FPN	**39.1**	**59.1**	**42.3**	**21.8**	**42.7**	50.2



	AP	time
[A] YOLOv2† [26]	21.6	25
[B] SSD321 [21]	28.0	61
[C] DSSD321 [9]	28.0	85
[D] R-FCN‡ [3]	29.9	85
[E] SSD513 [21]	31.2	125
[F] DSSD513 [9]	33.2	156
[G] FPN FRCN [19]	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	37.8	198

†Not plotted ‡Extrapolated time

RetinaNet在各个维度上去测量精度和速度，都明显比YOLO，SSD，DSSD有明显的进步

3

FCOS

Fully Convolutional One-Stage Object Detection

3 Fully Convolutional One-Stage Object Detection

问题动机

Anchor-based?

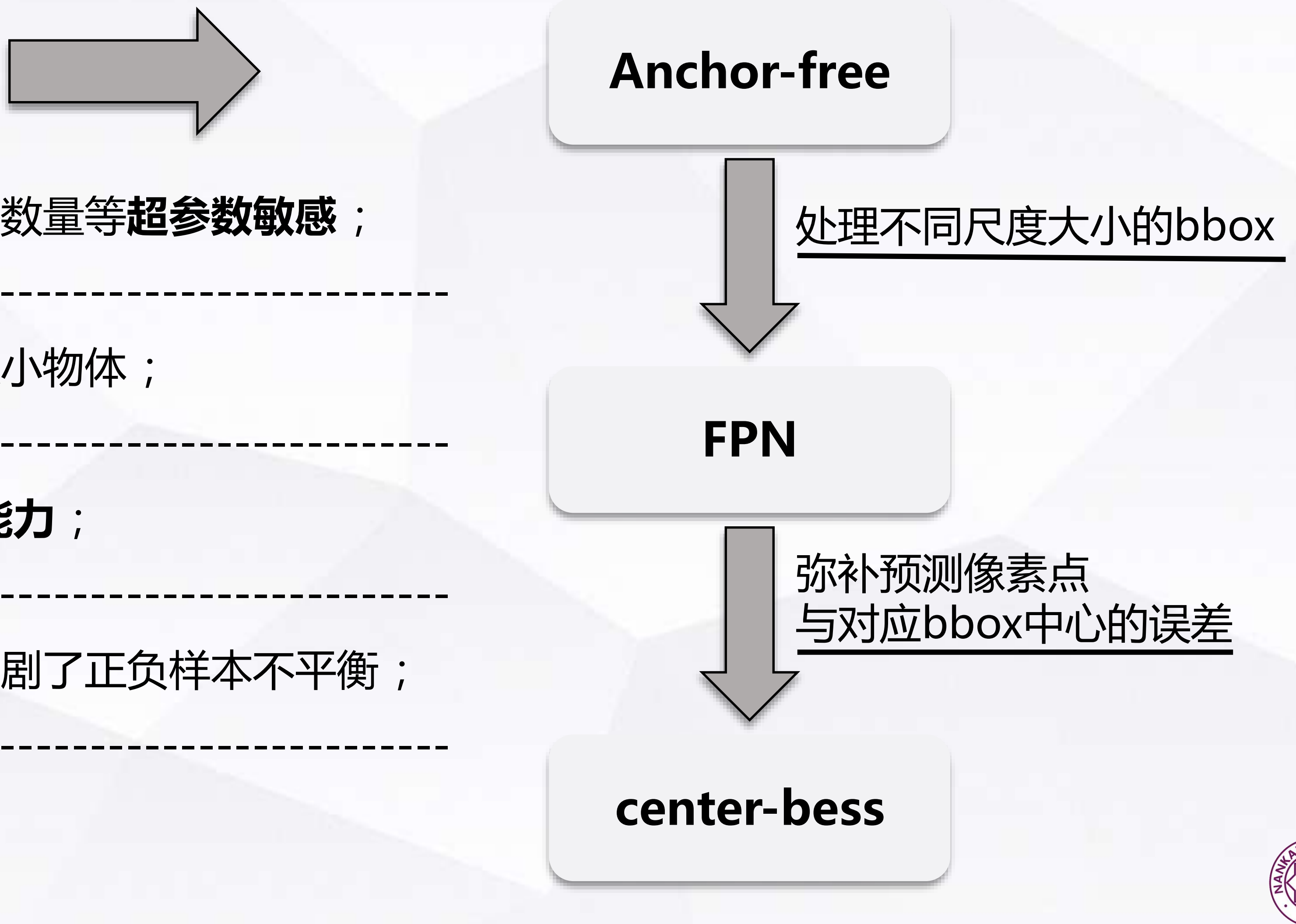
检测性能对锚点框的尺寸、长宽比和数量等**超参数敏感**；

难以处理形状变化大的物体，特别是小物体；

预设锚点框也限制了检测器的**泛化能力**；

为了高召回率**需放置大量锚点框**，加剧了正负样本不平衡；

锚点框涉及**复杂的重叠计算**。



Anchor-free

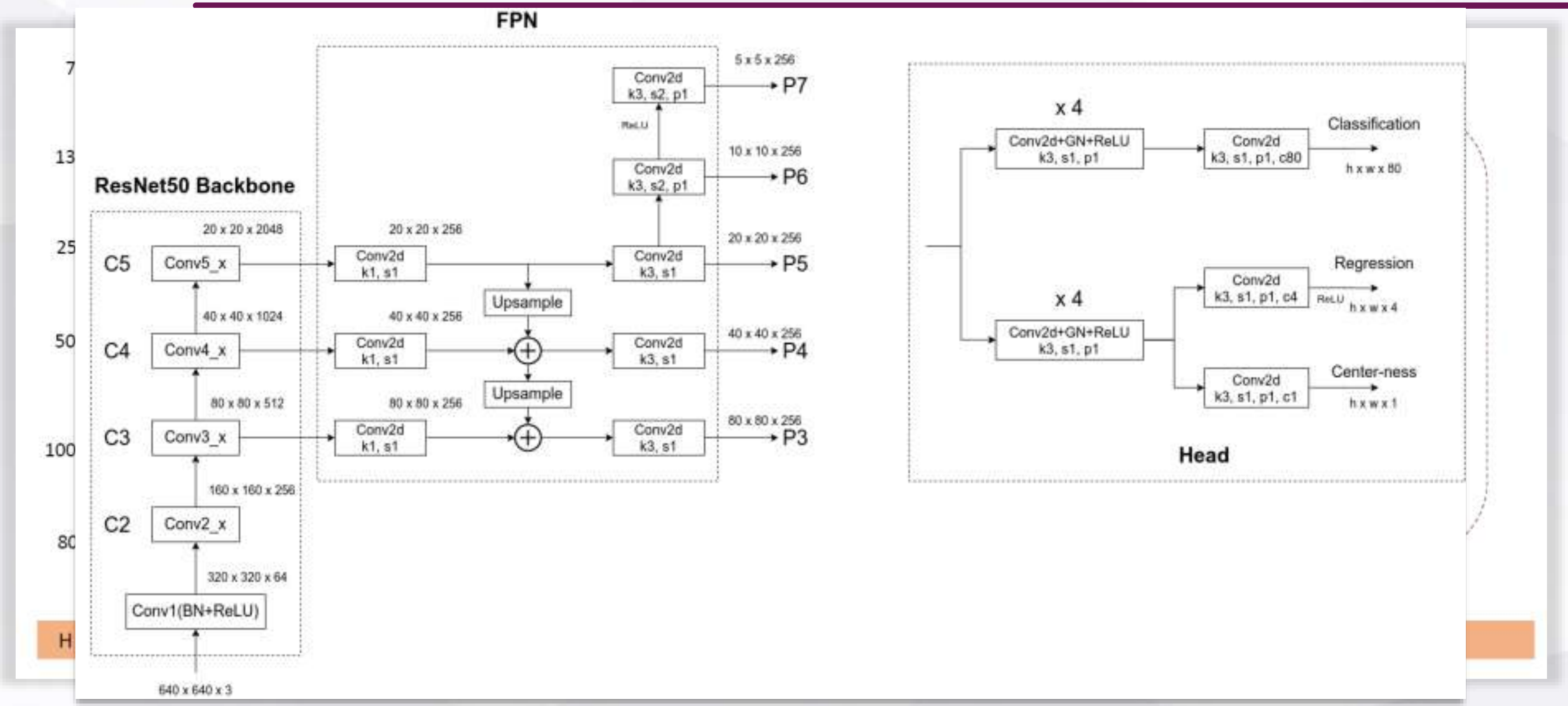
处理不同尺度大小的bbox

FPN

弥补预测像素点
与对应bbox中心的误差

center-bess

3 Fully Convolutional One-Stage Object Detection



输入端： 该网络的**输入图像大小**为 W 和 H 。
基准网络： 基准网络用来提取图片特征。论文使用**FPN网络**，对每个像素点进行多尺度预测。
Head输出端： Head用来完成**目标检测结果的输出。**
输出head有5个，这5个head共享权重，每个head有三个分支，分别为：
目标类别 (H×W×C)，中心度 (H×W×1)，和目标尺寸 (H×W×4)。

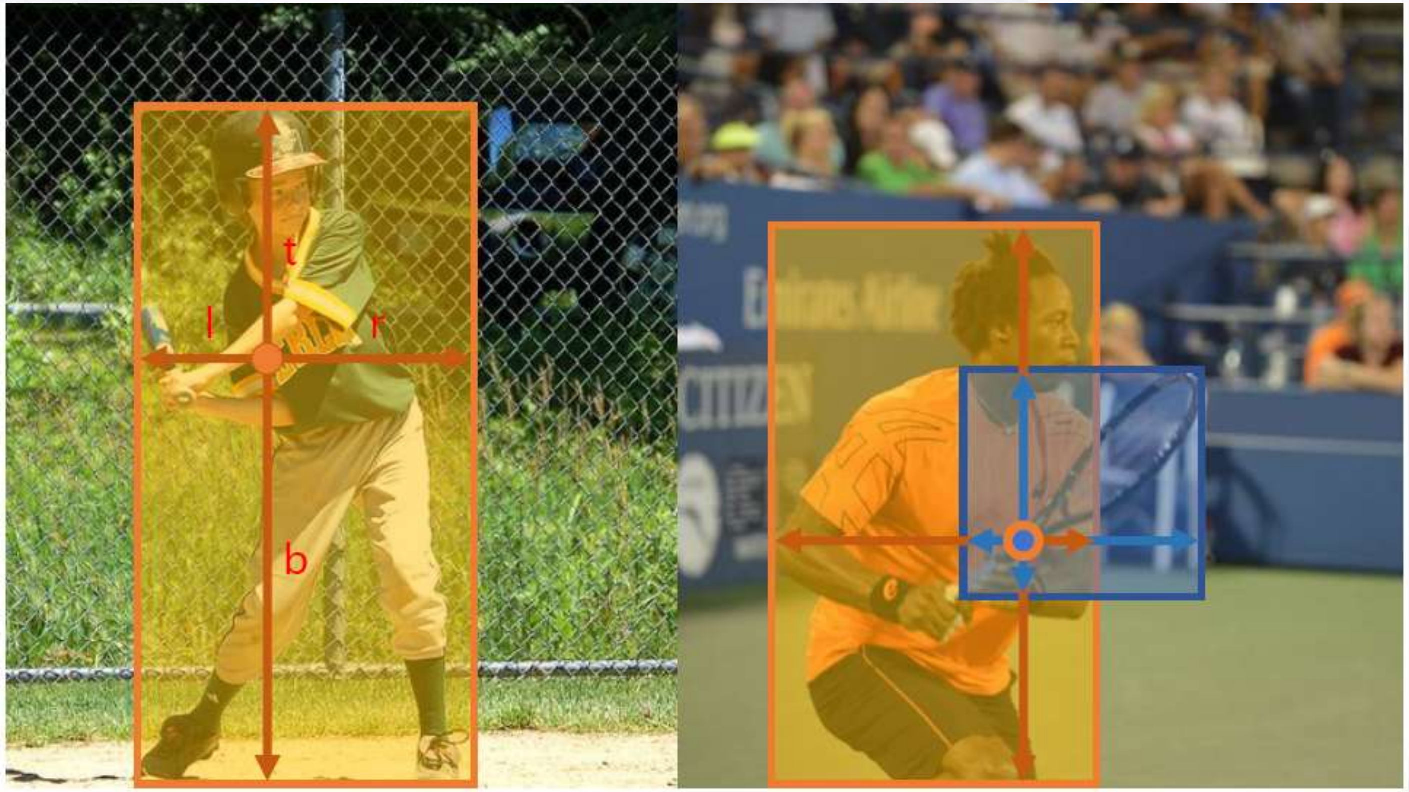
3 Fully Convolutional One-Stage Object Detection

如何预测bbox?

以右图左边人物目标为例，其bbox区域内所有黄色像素位置的类别都为person。bbox区域内的对应的特征图上每个像素点都有其对应的回归目标 (l*, r*, t*, b*) ，对于某个像素的坐标(x , y):

$$
l^* = x - x_0^{(i)}, \quad t^* = y - y_0^{(i)},
$$
$$
r^* = x_1^{(i)} - x, \quad b^* = y_1^{(i)} - y.
$$

上述回归目标为正数，所以网络输出时还需要进行指数运算，最后预测的bbox为(x , y , l , r , t , b)。

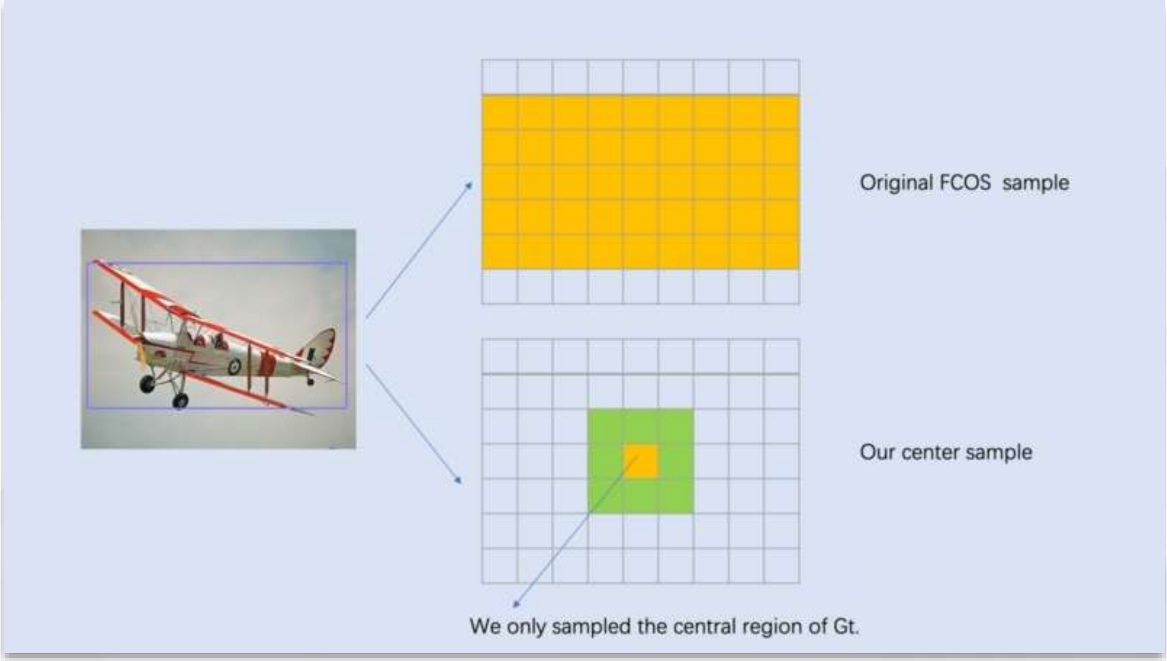


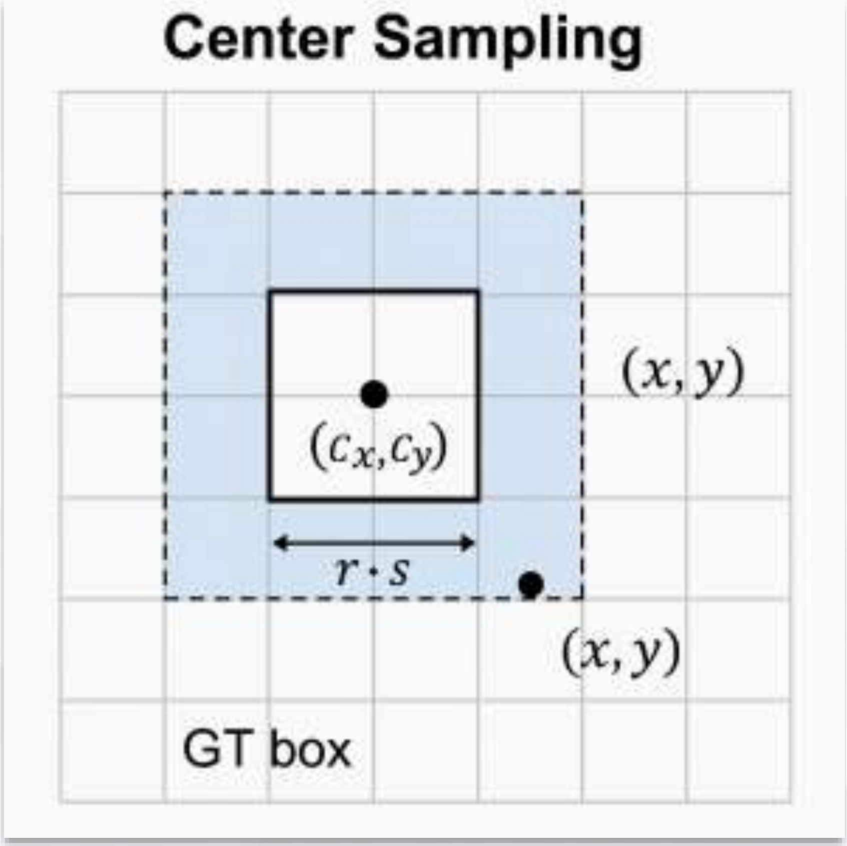
3 Fully Convolutional One-Stage Object Detection

后续完善更进——Center Sampling机制

原始论文：GT bbox 内的点，分类时均作为正样本（下图上面小图的所有黄色区域）。

改进 论文：只有 **GT bbox 中心附近的小 bbox**内的点，分类时才作为正样本。





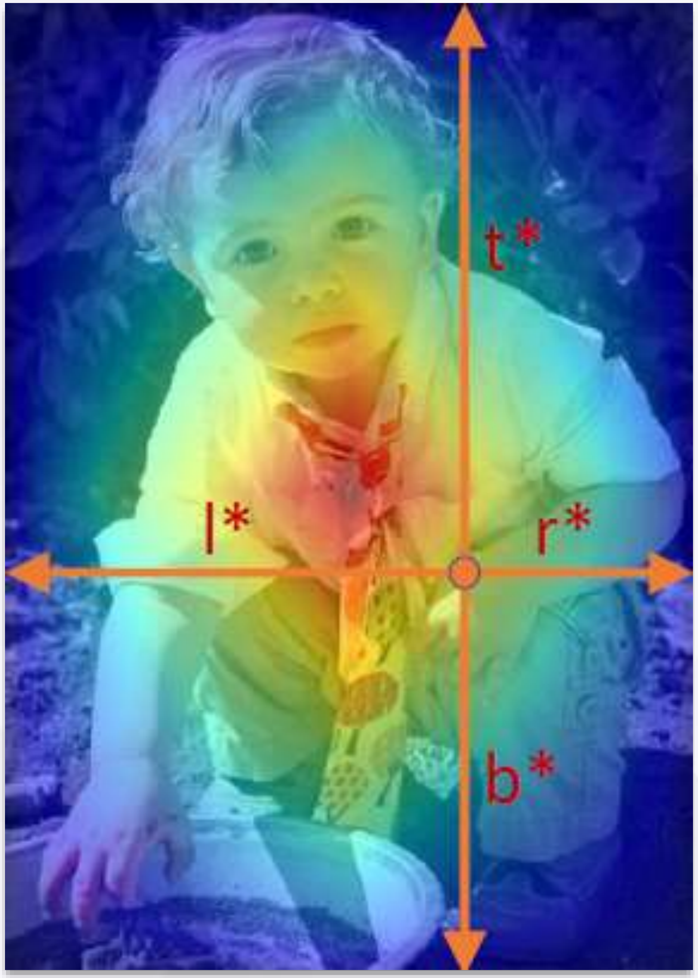
3 Fully Convolutional One-Stage Object Detection

什么是中心度?

$$l^* = x - x_0^{(i)}, \quad t^* = y - y_0^{(i)},$$
$$r^* = x_1^{(i)} - x, \quad b^* = y_1^{(i)} - y.$$

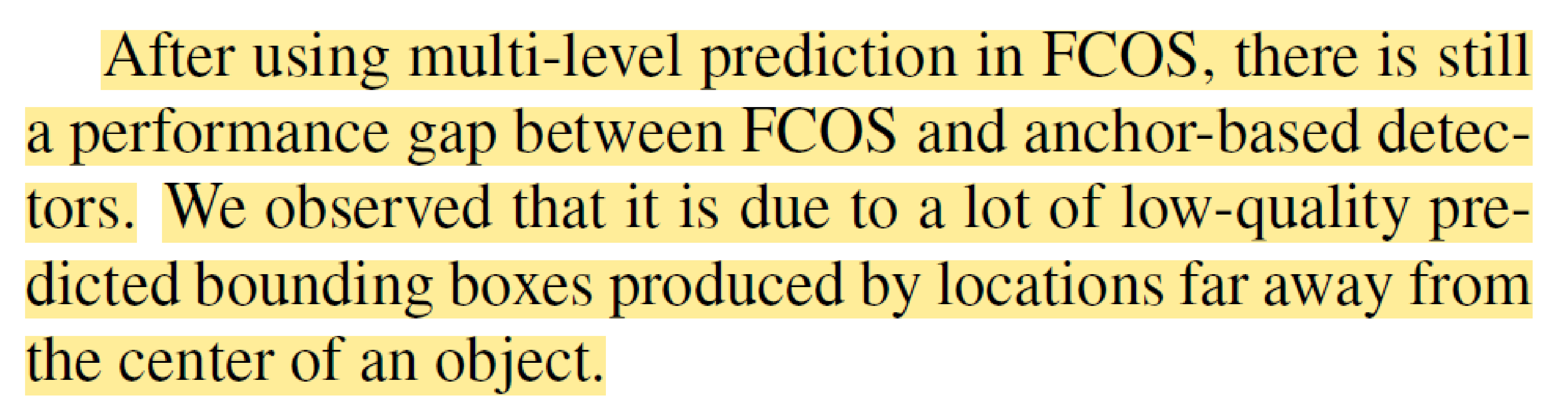
$$\text{centerness}^* = \sqrt{\frac{\min(l^*, r^*)}{\max(l^*, r^*)} \times \frac{\min(t^*, b^*)}{\max(t^*, b^*)}}$$

越靠近中心	l^***和**r^***,** t^*和 b^***越接近**	centerness* 越**接近1**	**中心度越大**
越靠近边缘	l^*和r^*, t^*和 b^*相差越大	centerness* 越接近0	中心度越小



3 Fully Convolutional One-Stage Object Detection

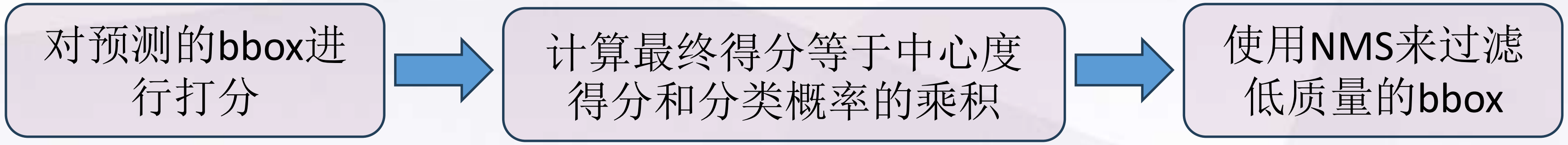
为什么要预测中心度?



After using multi-level prediction in FCOS, there is still a performance gap between FCOS and anchor-based detectors. We observed that it is due to a lot of low-quality predicted bounding boxes produced by locations far away from the center of an object.

如果不加这一项，FCOS性能会弱于anchor-based的检测模型，原因是模型会生成很多偏离目标中心的低质量bbox。

中心度的使用方法?



对预测的bbox进行打分

计算最终得分等于中心度得分和分类概率的乘积

使用NMS来过滤低质量的bbox

3 Fully Convolutional One-Stage Object Detection

损失函数

$$L(\{p_{x,y}\},\{t_{x,y}\}) = \frac{1}{N_{pos}}\sum_{x,y} L_{cls}(p_{x,y}, c^*_{x,y}) + \frac{\lambda}{N_{pos}}\sum_{x,y} \mathbb{I}_{\{c^*_{x,y}>0\}} L_{reg}(t_{x,y}, t^*_{x,y})$$

- •L_{cls} 是 Focal Loss， 用于分类任务。

- •L_{reg}是 IoU Loss (源自UnitBox)，用于边界框回归。

- •N_{pos} 是正样本的数量$c^*_{x,y}$ 。

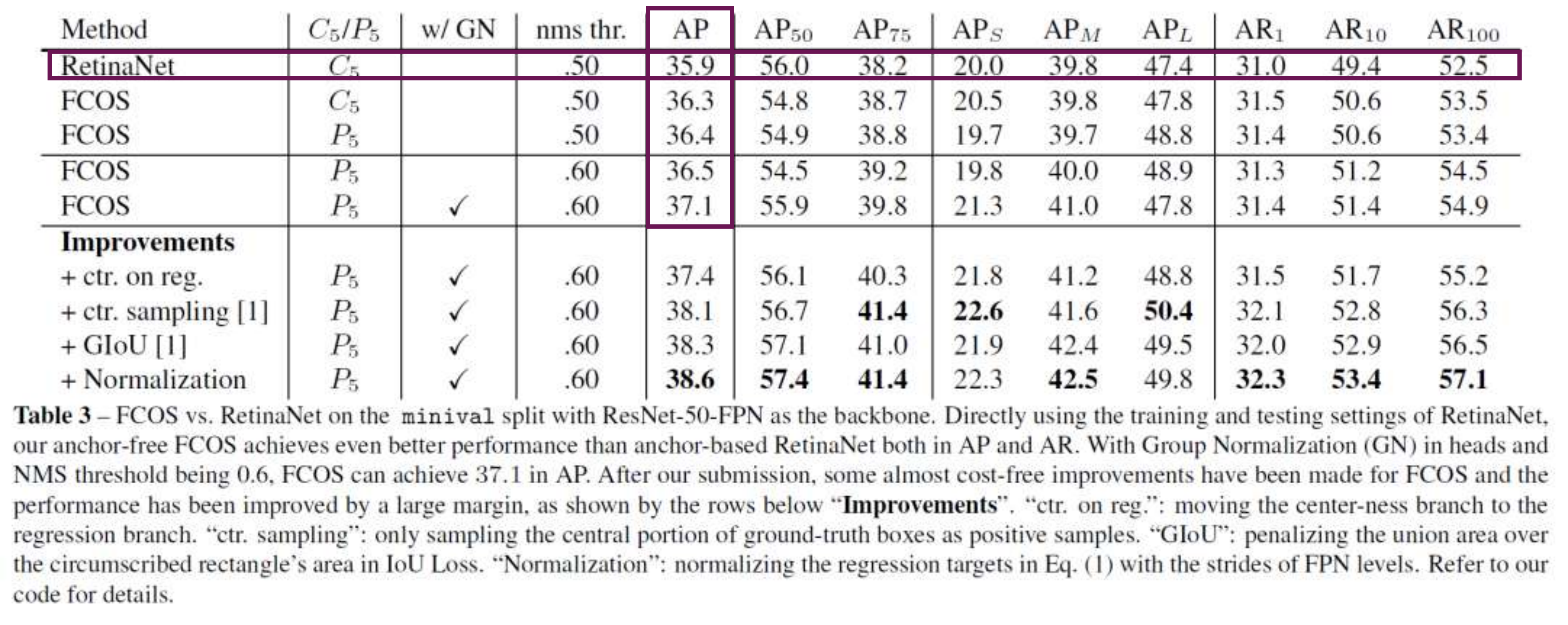
- •λ 是平衡 Lreg 权重的超参数，本文中设为1。

- •求和是在特征图 Fi 上的所有位置 (x,y) 进行的。

- •$\mathbb{I}_{\{c^*_{x,y}>0\}}$是指示函数，当 $c^*_{x,y}$ >0 (正样本) 时为1，否则为0。即只有正样本才贡献回归损失。

3 Fully Convolutional One-Stage Object Detection

anchor free设计本身的有效性

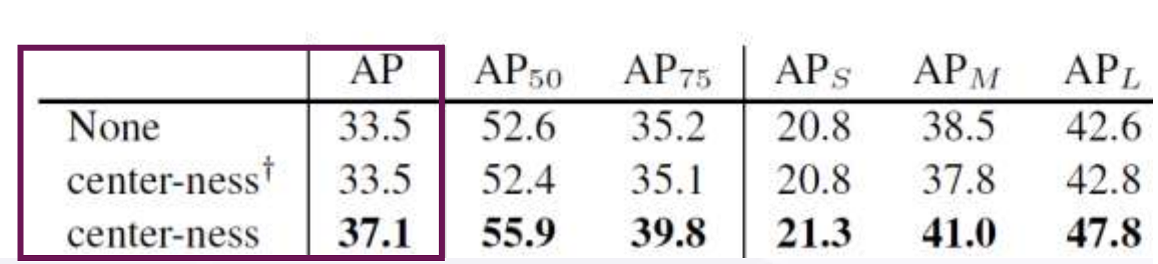


Method	C_5/P_5	w/ GN	nms thr.	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AR_1	AR_{10}	AR_{100}
RetinaNet	C_5		.50	35.9	56.0	38.2	20.0	39.8	47.4	31.0	49.4	52.5
FCOS	C_5		.50	36.3	54.8	38.7	20.5	39.8	47.8	31.5	50.6	53.5
FCOS	P_5		.50	36.4	54.9	38.8	19.7	39.7	48.8	31.4	50.6	53.4
FCOS	P_5		.60	36.5	54.5	39.2	19.8	40.0	48.9	31.3	51.2	54.5
FCOS	P_5	✓	.60	37.1	55.9	39.8	21.3	41.0	47.8	31.4	51.4	54.9
Improvements												
+ ctr. on reg.	P_5	✓	.60	37.4	56.1	40.3	21.8	41.2	48.8	31.5	51.7	55.2
+ ctr. sampling [1]	P_5	✓	.60	38.1	56.7	**41.4**	**22.6**	41.6	**50.4**	32.1	52.8	56.3
+ GIoU [1]	P_5	✓	.60	38.3	57.1	41.0	21.9	42.4	49.5	32.0	52.9	56.5
+ Normalization	P_5	✓	.60	**38.6**	**57.4**	**41.4**	22.3	**42.5**	49.8	**32.3**	**53.4**	**57.1**

Table 3 – FCOS vs. RetinaNet on the `minival` split with ResNet-50-FPN as the backbone. Directly using the training and testing settings of RetinaNet, our anchor-free FCOS achieves even better performance than anchor-based RetinaNet both in AP and AR. With Group Normalization (GN) in heads and NMS threshold being 0.6, FCOS can achieve 37.1 in AP. After our submission, some almost cost-free improvements have been made for FCOS and the performance has been improved by a large margin, as shown by the rows below "**Improvements**". "ctr. on reg.": moving the center-ness branch to the regression branch. "ctr. sampling": only sampling the central portion of ground-truth boxes as positive samples. "GIoU": penalizing the union area over the circumscribed rectangle's area in IoU Loss. "Normalization": normalizing the regression targets in Eq. (1) with the strides of FPN levels. Refer to our code for details.

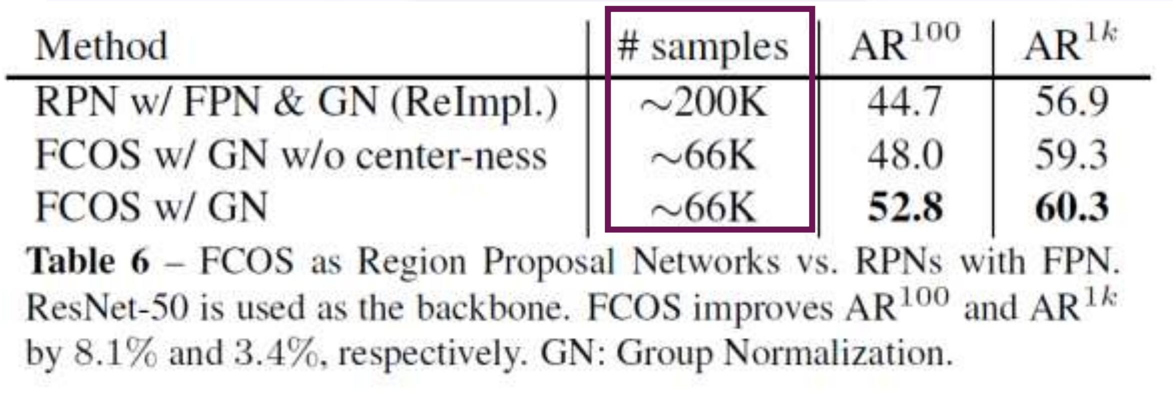
3 Fully Convolutional One-Stage Object Detection

验证Center—ness模块的贡献



	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
None	33.5	52.6	35.2	20.8	38.5	42.6
center-ness[†]	33.5	52.4	35.1	20.8	37.8	42.8
center-ness	**37.1**	**55.9**	**39.8**	**21.3**	**41.0**	**47.8**

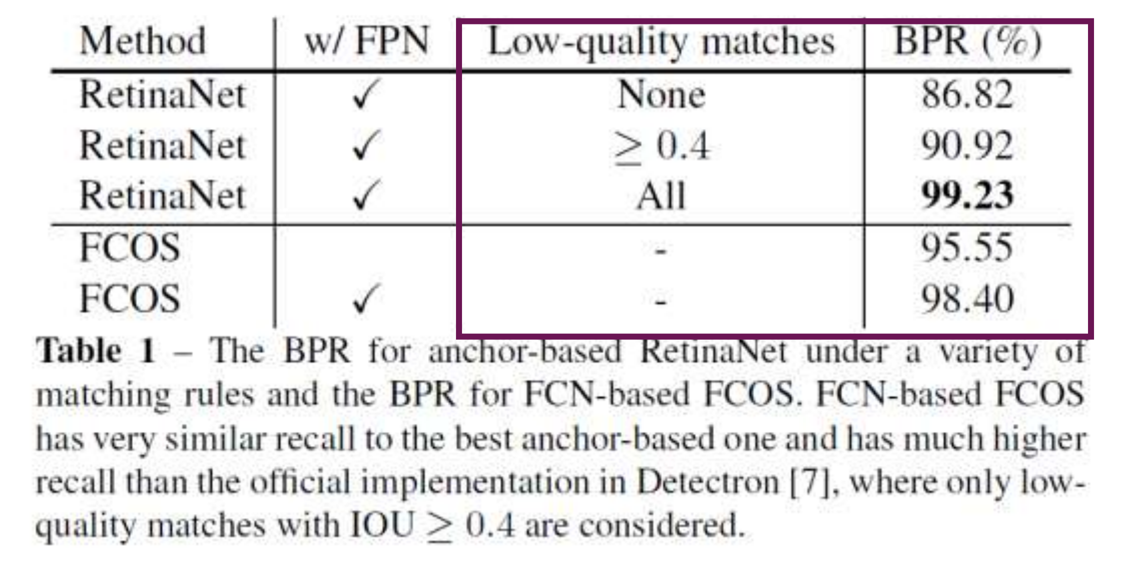
验证anchor free检测器推理的速度优势



Method	# samples	AR^{100}	AR^{1k}
RPN w/ FPN & GN (ReImpl.)	∼200K	44.7	56.9
FCOS w/ GN w/o center-ness	∼66K	48.0	59.3
FCOS w/ GN	∼66K	**52.8**	**60.3**

Table 6 – FCOS as Region Proposal Networks vs. RPNs with FPN. ResNet-50 is used as the backbone. FCOS improves AR^{100} and AR^{1k} by 8.1% and 3.4%, respectively. GN: Group Normalization.

验证FPN对FCOS的效果



Method	w/ FPN	Low-quality matches	BPR (%)
RetinaNet	✓	None	86.82
RetinaNet	✓	≥ 0.4	90.92
RetinaNet	✓	All	**99.23**
FCOS		-	95.55
FCOS	✓	-	98.40

Table 1 – The BPR for anchor-based RetinaNet under a variety of matching rules and the BPR for FCN-based FCOS. FCN-based FCOS has very similar recall to the best anchor-based one and has much higher recall than the official implementation in Detectron [7], where only low-quality matches with IOU ≥ 0.4 are considered.

将FPN融入FCOS去检测小目标，中目标和大目标，然后与第二个实验中baseline进行对比，结果显示这些都有显著提高，尤其是小目标更加精细化的比baseline会有显著提升

4

Generalized Focal Loss

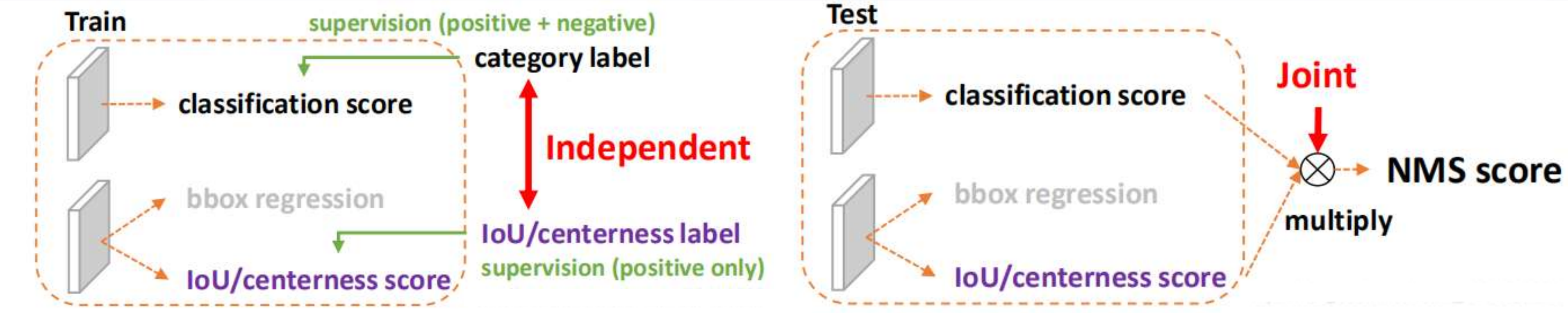
Learning Qualified and Distributed Bounding Boxes for Dense Object Detection

4 Generalized Focal Loss

存在问题一

classification score 和 IoU/centerness score 训练测试不一致

1 用法不一致



2 对象不一致

4 Generalized Focal Loss

存在问题二

bbox regression 采用的表示不够灵活，无法表示现实中具有很强的不确定性边界框的

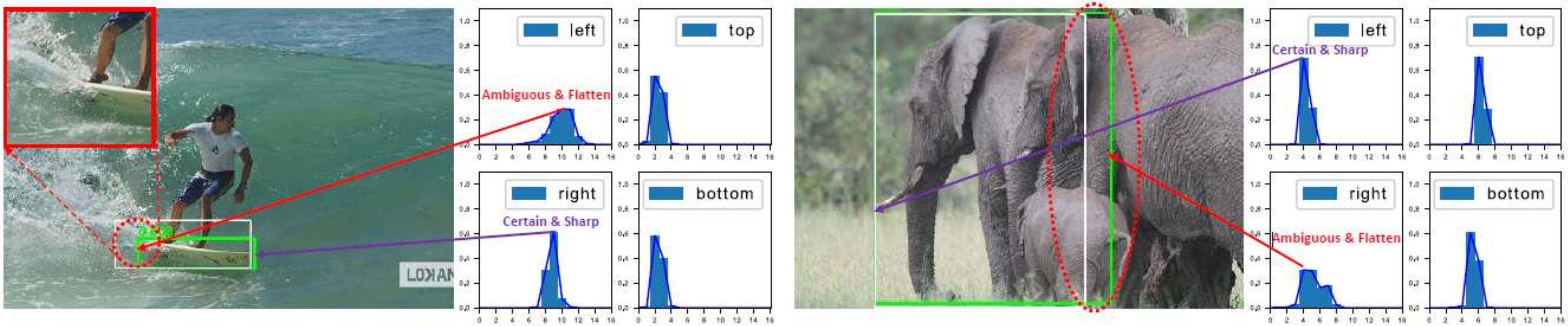
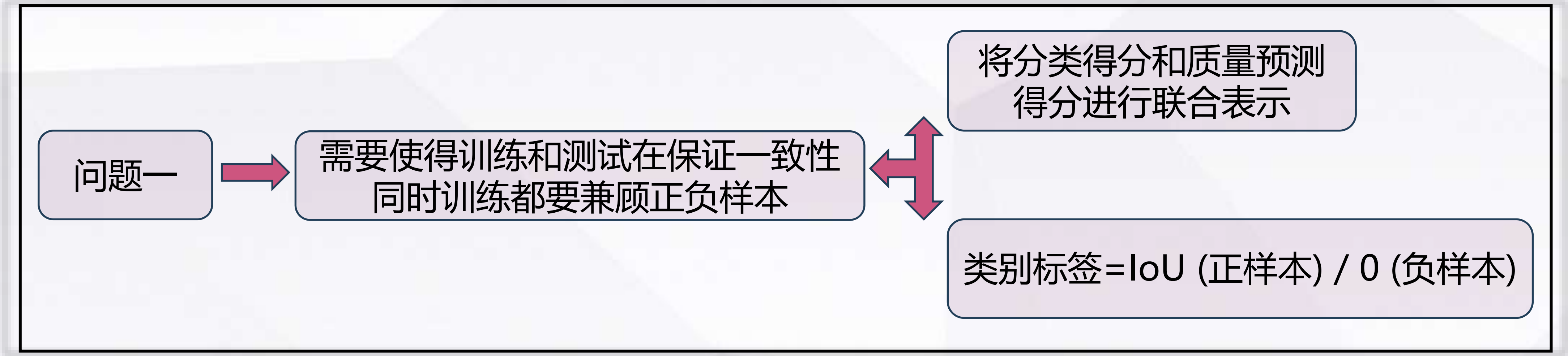


Figure 3: Due to occlusion, shadow, blur, etc., the boundaries of many objects are not clear enough, so that the ground-truth labels (white boxes) are sometimes not credible and Dirac delta distribution is limited to indicate such issues. Instead, the proposed learned representation of General distribution for bounding boxes can reflect the underlying information by its shape, where a flatten distribution denotes the unclear and ambiguous boundaries (see red circles) and a sharp one stands for the clear cases. The predicted boxes by our model are marked green.

4 Generalized Focal Loss

解决方法

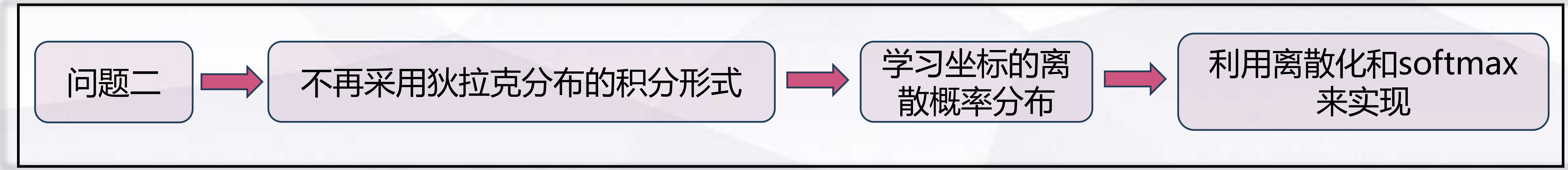


问题一

需要使得训练和测试在保证一致性同时训练都要兼顾正负样本

将分类得分和质量预测得分进行联合表示

类别标签=IoU (正样本) / 0 (负样本)



问题二

不再采用狄拉克分布的积分形式

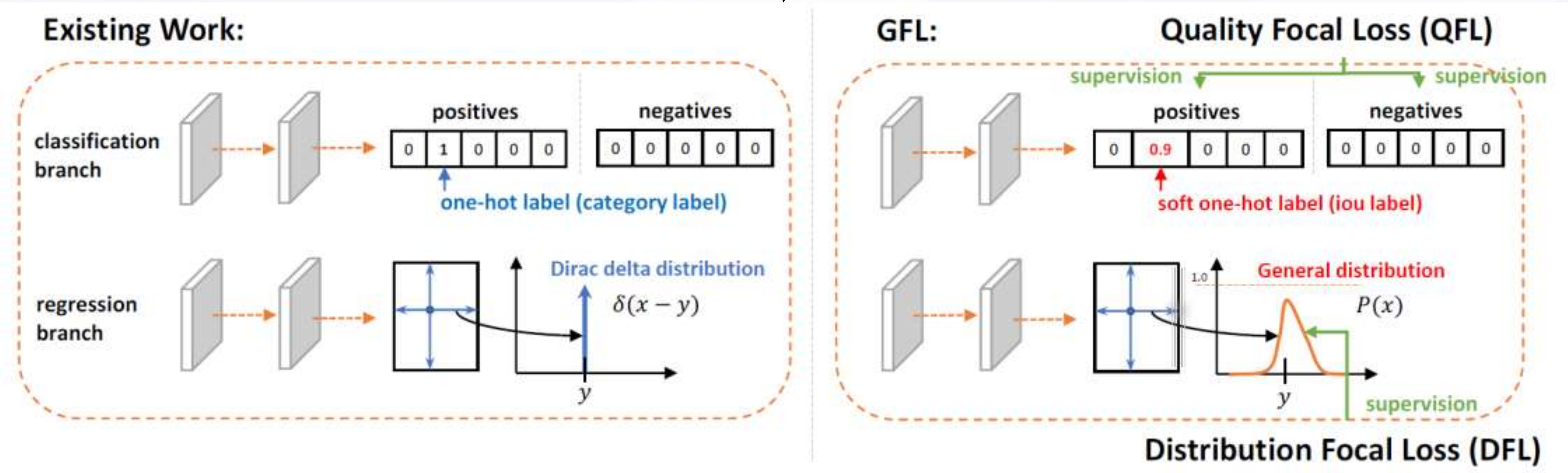
学习坐标的离散概率分布

利用离散化和softmax来实现

4 Generalized Focal Loss

$$\mathrm{FL}(p_t) = -(1-p_t)^{\gamma}\log(p_t) \quad p_t = \begin{cases} p & y = 1 \\ 1-p & y \neq 1 \end{cases}$$

这里y的label是离散的（非0即1）

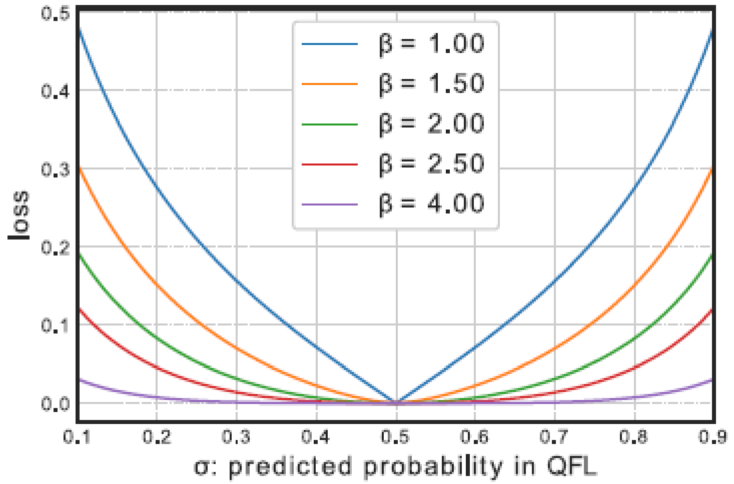


4 Generalized Focal Loss

针对第一个问题：QFL
由于之前的FL只能处理one-hot也就是离散的情况，在转成soft-one-hot之后，就需要对现有的FL进行修改：
修改后的FL既要保留原有FL对于类别不均衡的优势，又要能够处理连续的label值，因此，对FL的扩展表现在两方面：

	FL	QFL
交叉熵部分	$(1-P_t)^{\gamma}$	$\lvert y-\sigma\rvert^{\beta} \quad (\beta \geq 0)$
每个样本的缩放因子	$-\log(p_t)$	$-\big((1-\mathrm{y})\log(1-\sigma)+\mathrm{y}\log(\sigma)\big)$
完整公式	$\mathrm{QFL}=-\lvert y-\sigma\rvert^{\beta}\big((1-\mathrm{y})\log(1-\sigma)+\mathrm{y}\log(\sigma)\big)$	

在y=0.5的时候，不同β的可视化的图，在文中使用β=2：



4 Generalized Focal Loss

针对第二个问题：DFL
由于真实的分布通常不会距离标注的位置太远，所以作者添加了DFL，希望网络能够快速地聚焦到标注位置附近的数值。

	公式推演发展
当前坐标点到4条边的距离	$y = \int_{-\infty}^{+\infty} \delta(x - y)x\,\mathrm{d}x$
常规操作时将y作为狄拉克分布来回归	$\hat{y} = \int_{-\infty}^{+\infty} \mathrm{P}(x)x\,\mathrm{d}x = \int_{y_0}^{y_n} P(x)xdx$
改成离散形式	$\hat{y} = \sum_{i=0}^{n} P(yi)y_i$
提出DFL	$DFL(S_i, S_i + 1) = -\left((y_{i+1} - y)\log(\boldsymbol{S}_i) + (y - y_i)\log(S_{i+1})\right)$

4 Generalized Focal Loss

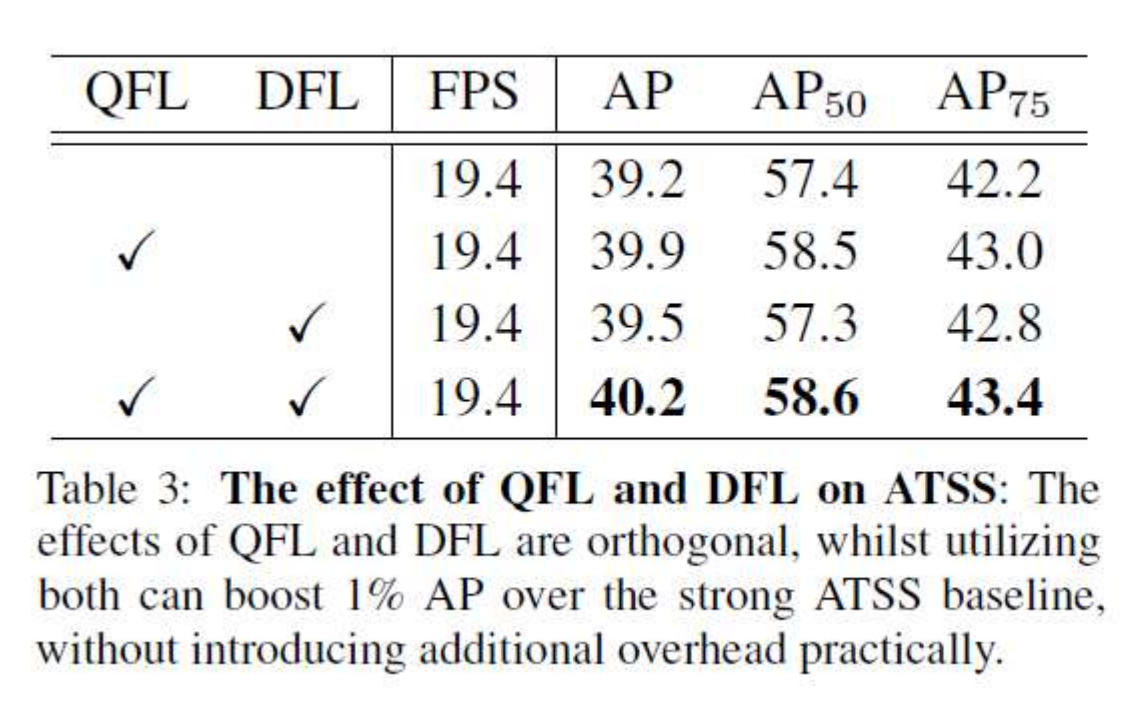
将QFL和DFL相结合得到GFL

$$GFL(p_{yl}, p_{y_r}) = -\left|y - (y_l p_{yl} + y_\gamma p_{yr})\right|^{\beta} \left((y_r - y)\log(p_{yl}) + (y - y_l)\log(p_{yr})\right)$$

使用GFL来训练，使用loss为

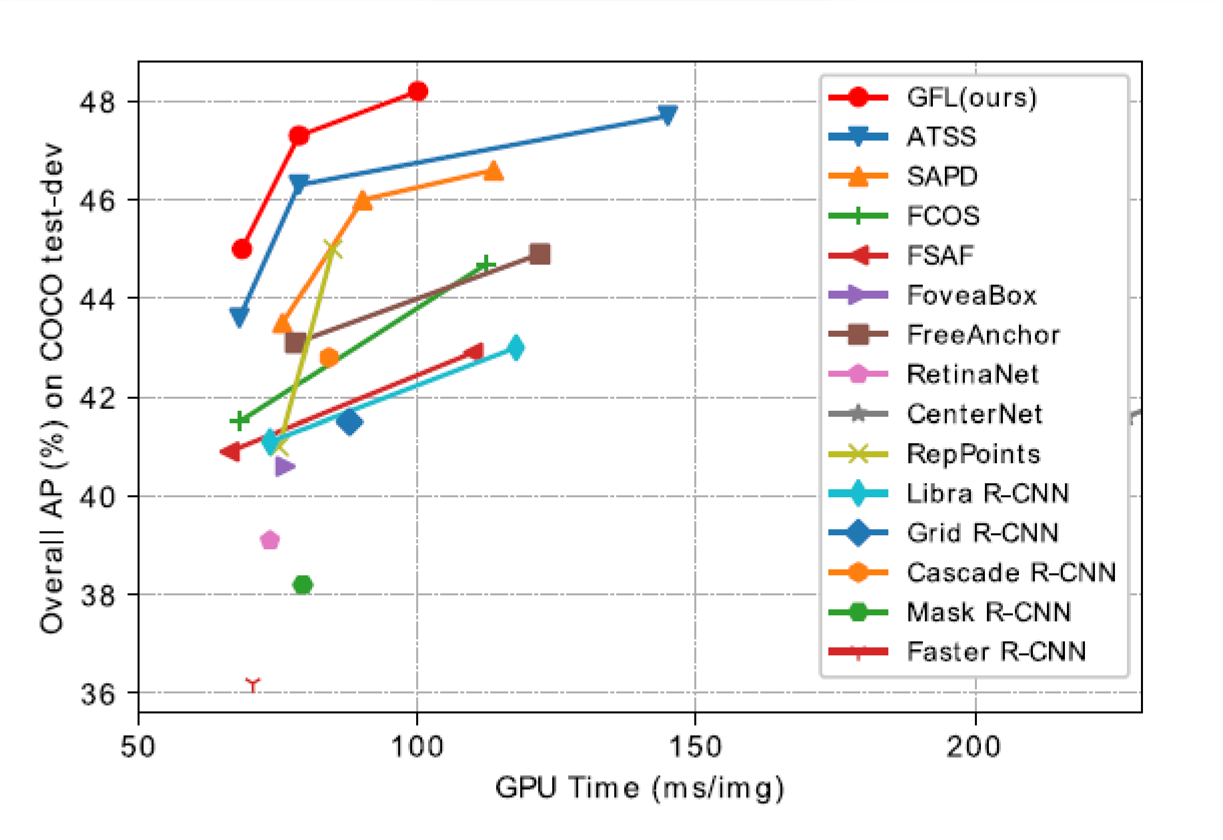
$$\mathcal{L} = \frac{1}{N_{pos}} \sum_z \underbrace{\boxed{\mathcal{L}_Q}}_{\text{QFL}} + \frac{1}{N_{pos}} \sum_z \mathbf{1}_{\{c_z^* > 0\}} \left(\lambda_0 \underbrace{\boxed{\mathcal{L}_B}}_{\text{GIoU Loss}} + \lambda_1 \underbrace{\boxed{\mathcal{L}_D}}_{\text{DFL}}\right)$$

4 Generalized Focal Loss



QFL	DFL	FPS	AP	AP$_{50}$	AP$_{75}$
		19.4	39.2	57.4	42.2
✓		19.4	39.9	58.5	43.0
	✓	19.4	39.5	57.3	42.8
✓	✓	19.4	**40.2**	**58.6**	**43.4**

Table 3: **The effect of QFL and DFL on ATSS**: The effects of QFL and DFL are orthogonal, whilst utilizing both can boost 1% AP over the strong ATSS baseline, without introducing additional overhead practically.



5

Generalized Focal Loss V2

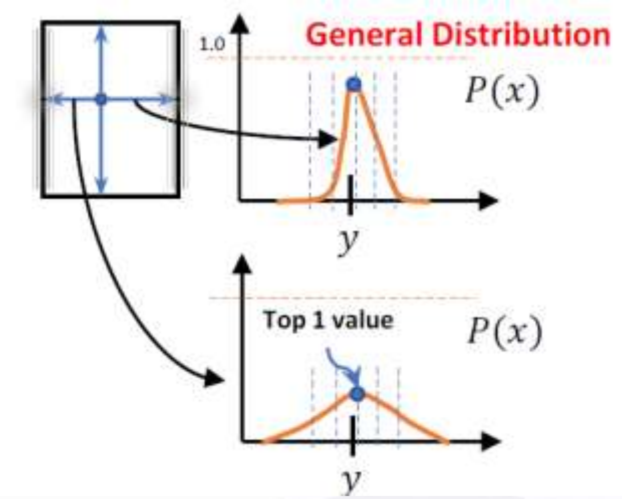
Learning Reliable Localization Quality Estimation for Dense Object Detection

5 Generalized Focal Loss V2

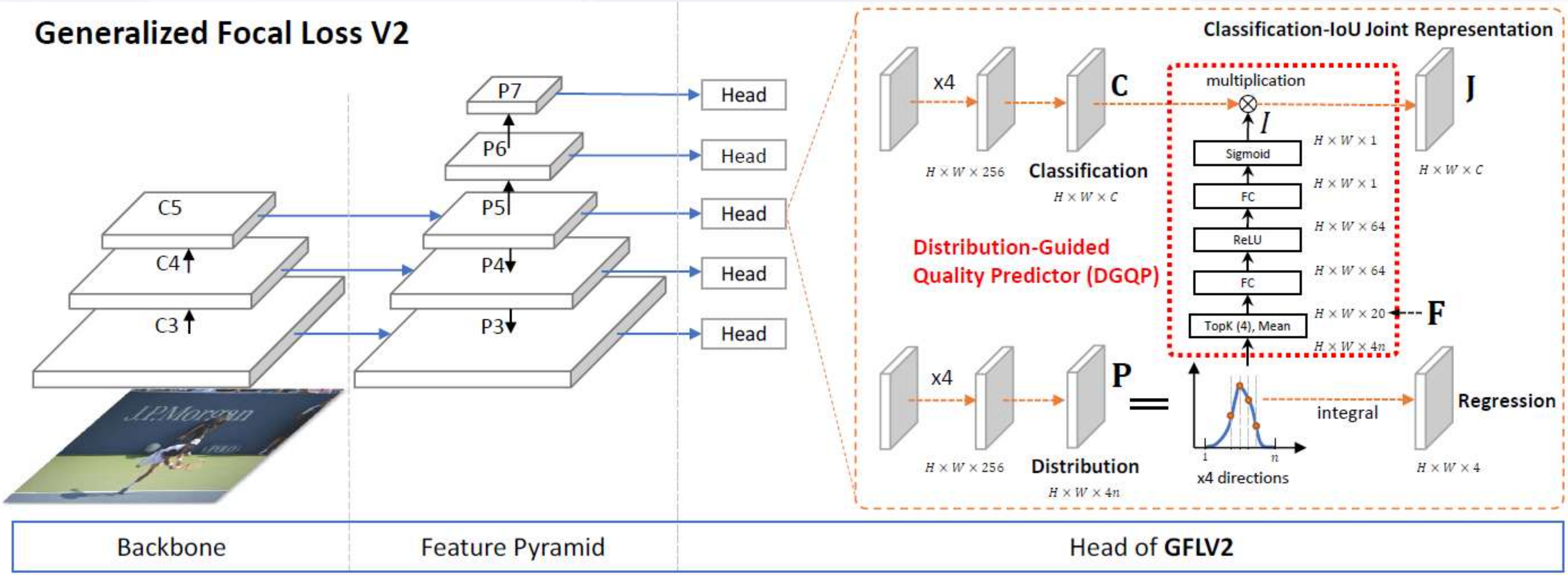
来自GFVL1启发

GFLV1中作者团队提出了**General Distribution**，来定义更一般化的位置分布，而且发现了**边界越模糊，分布越平缓。**

GFLV2利用这个**分布信息**去指导**最终定位质量的估计。**



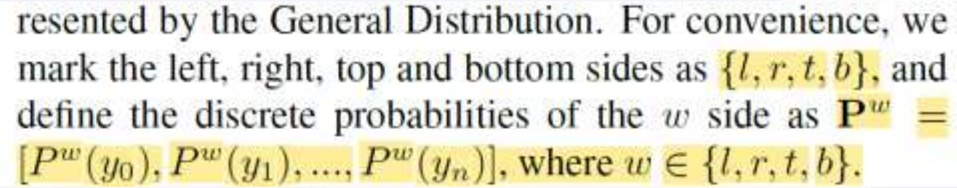
5 Generalized Focal Loss V2



精髓之处：DGQP

那么DGQP（分布引导的质量预测器）具体是什么呢?

作者将中心点到四个框的距离不再是用一个具体的数值来表示，而是将每一个量使用一系列概率来表示



resented by the General Distribution. For convenience, we mark the left, right, top and bottom sides as $\{l, r, t, b\}$, and define the discrete probabilities of the w side as $\mathbf{P}^w = [P^w(y_0), P^w(y_1), ..., P^w(y_n)]$, where $w \in \{l, r, t, b\}$.

举个例子：当ω=l时，可能的取值有个值，这个P就可以用来表示每个取值的概率

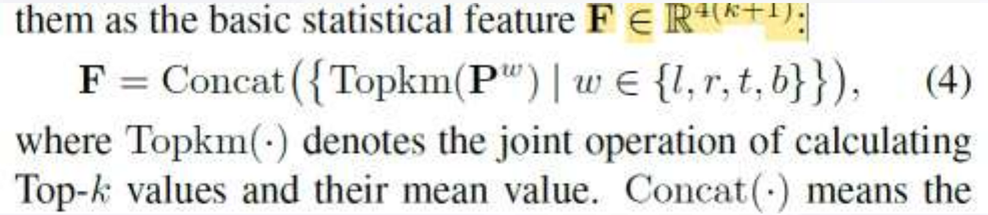
$$P^l = \left[\mathrm{P}^l_{(y_0)}, [\mathrm{P}^l_{(y_1)},], \ldots, [\mathrm{P}^l_{(y_5)},]\right] = [0.02, 0.05, 0.08, 0.6, 0.2, 0.05]$$

由上述例子我们可以看出P^l分布峰值明显比较尖锐的时候，说明峰值取值更加可信，而分布越平坦，可信度度越低

5 Generalized Focal Loss V2

接着上个例子：$P^l = [0.02, 0.05, 0.08, 0.6, 0.2, 0.05]$，取Top-2=[0.6，0.2]，再取他们的平均值mean=0.4，在上下左右四个方向上都进行这样的操作，一共由产生12个数，构成特征向量$F \in \mathbb{R}^{12}$

从特殊到一般，对bounding box预测四个边的分布，分别取出Top-k和他们的均值，再进行Concat得到4（k+1）个值


them as the basic statistical feature $\mathbf{F} \in \mathbb{R}^{4(k+1)}$:

$$\mathbf{F} = \mathrm{Concat}(\{\mathrm{Topkm}(\mathbf{P}^w) \mid w \in \{l, r, t, b\}\}), \quad (4)$$

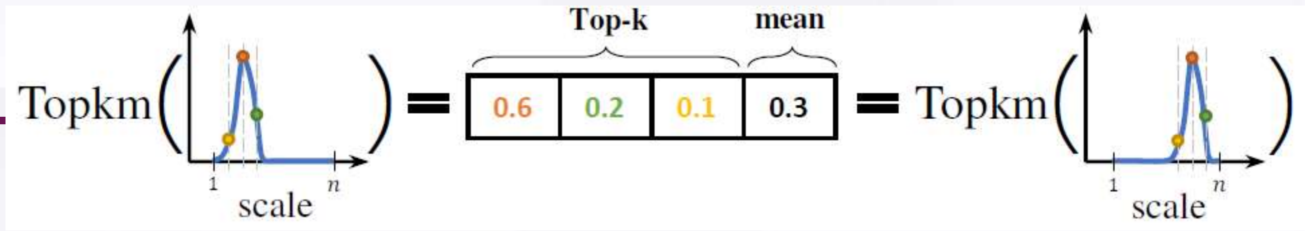
where $\mathrm{Topkm}(\cdot)$ denotes the joint operation of calculating Top-k values and their mean value. $\mathrm{Concat}(\cdot)$ means the

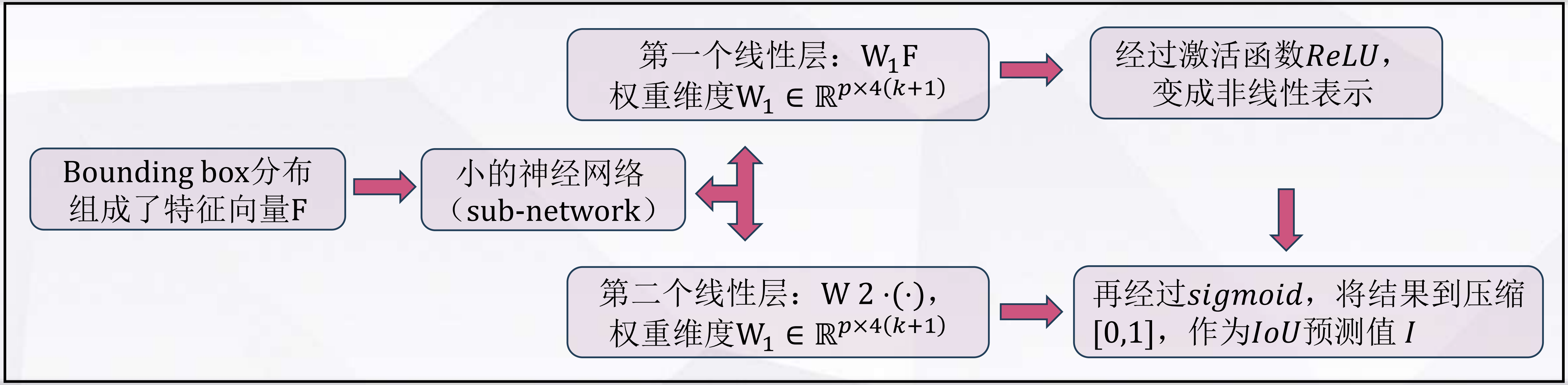
为什么要Top-k和mean呢?

Top-k 越大、mean 越集中	分布越尖锐	网络预测有信心	也就是说，分布的尖 或平，可以通过 top-k + mean 表征出来
Top-k 很平均、mean 值也不高	分布越平缓	网络不确定	
当像素位置发生平移变化	分布形状不会改变	Top-k和mean不变	平移不变性

4 Generalized Focal Loss V2

解决方法





Bounding box分布组成了特征向量F

小的神经网络（sub-network）

第一个线性层：W_1F
权重维度$W_1 \in \mathbb{R}^{p\times 4(k+1)}$

经过激活函数*ReLU*，变成非线性表示

第二个线性层：W 2 ·(·)，
权重维度$W_1 \in \mathbb{R}^{p\times 4(k+1)}$

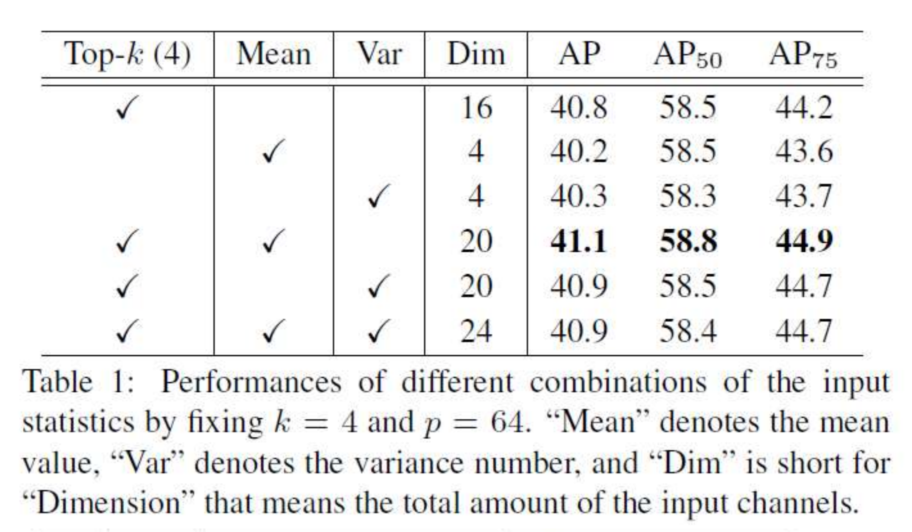
再经过*sigmoid*，将结果到压缩[0,1]，作为*IoU*预测值 *I*

接着上个例子：假设F=$\begin{bmatrix} 0.6 & 0.2 & 0.4 \\ 0.9 & 0.8 & 0.85 \\ 0.88 & 0.82 & 0.85 \\ 0.83 & 0.81 & 082 \end{bmatrix}$，经过$W_1$F每一行都加权平均，$z_1$=$\begin{bmatrix} 0.1 \times sum(F) \\ 0.2 \times sum(F) \\ 0.3 \times sum(F) \\ 0.4 \times sum(F) \end{bmatrix}$,再将$z_1$每一行的值分权相加得到$z_2$一维的值，最后经过*sigmoid*得到一个IoU分数，越靠近1说明这个框被预测的很准

5 Generalized Focal Loss V2

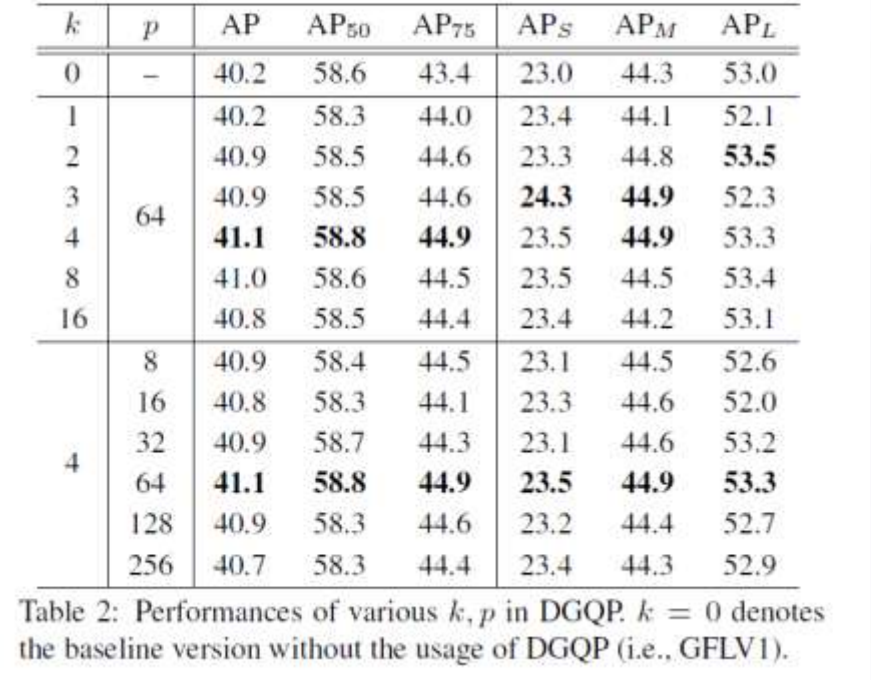
在Top-k(k=4)以及DGQP隐层单元为64的设置下，验证Top-k值，mean以及var对AP值的影响

DGQP(即k，p)的结构，也就是k和p取不同值对于最终结果的影响



Top-k (4)	Mean	Var	Dim	AP	AP$_{50}$	AP$_{75}$
✓			16	40.8	58.5	44.2
	✓		4	40.2	58.5	43.6
		✓	4	40.3	58.3	43.7
✓	✓		20	**41.1**	**58.8**	**44.9**
✓		✓	20	40.9	58.5	44.7
✓	✓	✓	24	40.9	58.4	44.7

Table 1: Performances of different combinations of the input statistics by fixing $k = 4$ and $p = 64$. “Mean” denotes the mean value, “Var” denotes the variance number, and “Dim” is short for “Dimension” that means the total amount of the input channels.



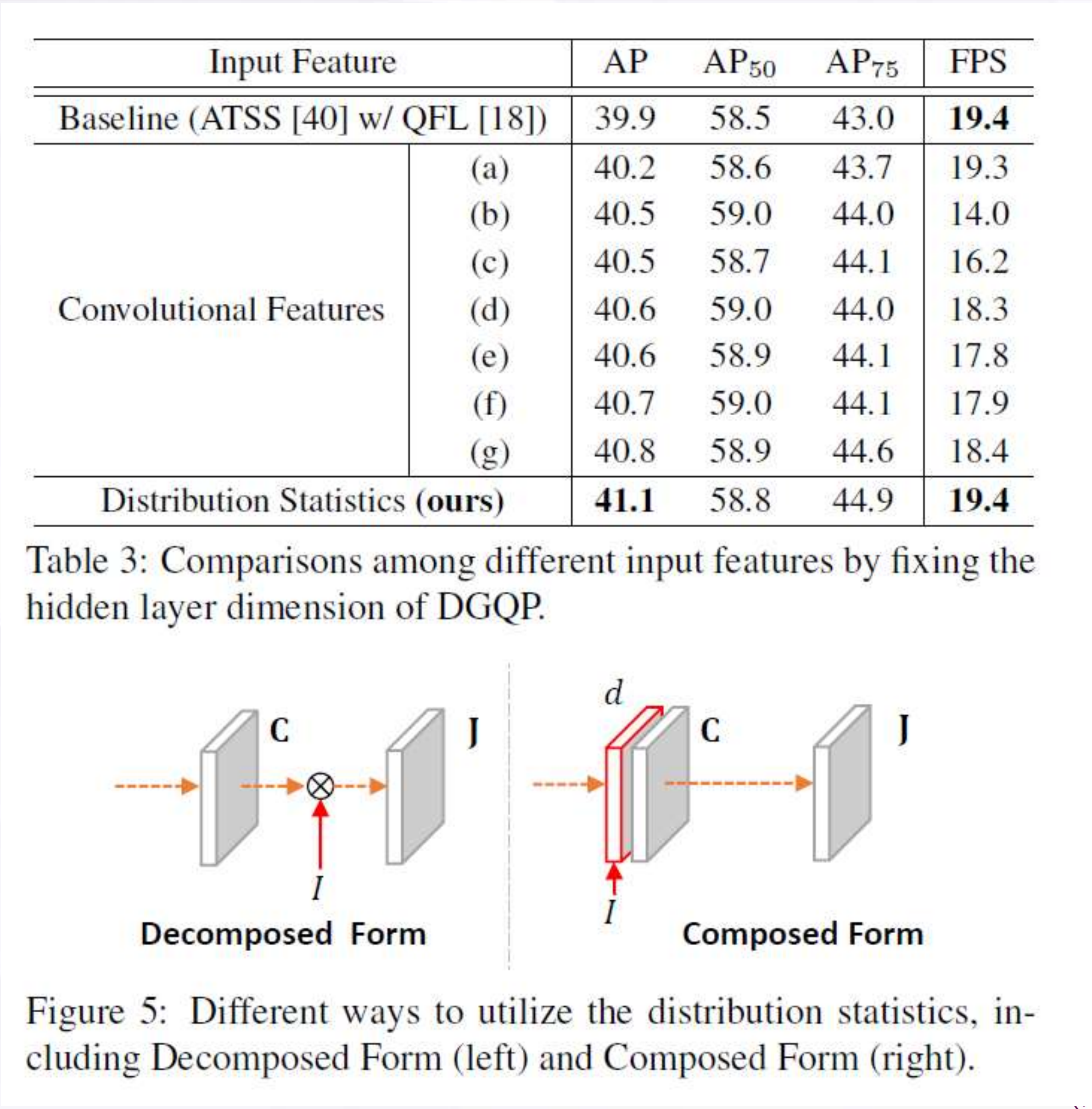
k	p	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$
0	–	40.2	58.6	43.4	23.0	44.3	53.0
1	64	40.2	58.3	44.0	23.4	44.1	52.1
2		40.9	58.5	44.6	23.3	44.8	**53.5**
3		40.9	58.5	44.6	**24.3**	**44.9**	52.3
4		**41.1**	**58.8**	**44.9**	23.5	**44.9**	53.3
8		41.0	58.6	44.5	23.5	44.5	53.4
16		40.8	58.5	44.4	23.4	44.2	53.1
4	8	40.9	58.4	44.5	23.1	44.5	52.6
	16	40.8	58.3	44.1	23.3	44.6	52.0
	32	40.9	58.7	44.3	23.1	44.6	53.2
	64	**41.1**	**58.8**	**44.9**	**23.5**	**44.9**	**53.3**
	128	40.9	58.3	44.6	23.2	44.4	52.7
	256	40.7	58.3	44.4	23.4	44.3	52.9

Table 2: Performances of various k, p in DGQP. $k = 0$ denotes the baseline version without the usage of DGQP (i.e., GFLV1).

5 Generalized Focal Loss V2

和使用基于卷积特征进行Quality Estimation 的进行对比

组合分类score C和质量估计J的两种形式



Input Feature		AP	AP_{50}	AP_{75}	FPS
Baseline (ATSS [40] w/ QFL [18])		39.9	58.5	43.0	**19.4**
Convolutional Features	(a)	40.2	58.6	43.7	19.3
	(b)	40.5	59.0	44.0	14.0
	(c)	40.5	58.7	44.1	16.2
	(d)	40.6	59.0	44.0	18.3
	(e)	40.6	58.9	44.1	17.8
	(f)	40.7	59.0	44.1	17.9
	(g)	40.8	58.9	44.6	18.4
Distribution Statistics **(ours)**		**41.1**	58.8	44.9	**19.4**

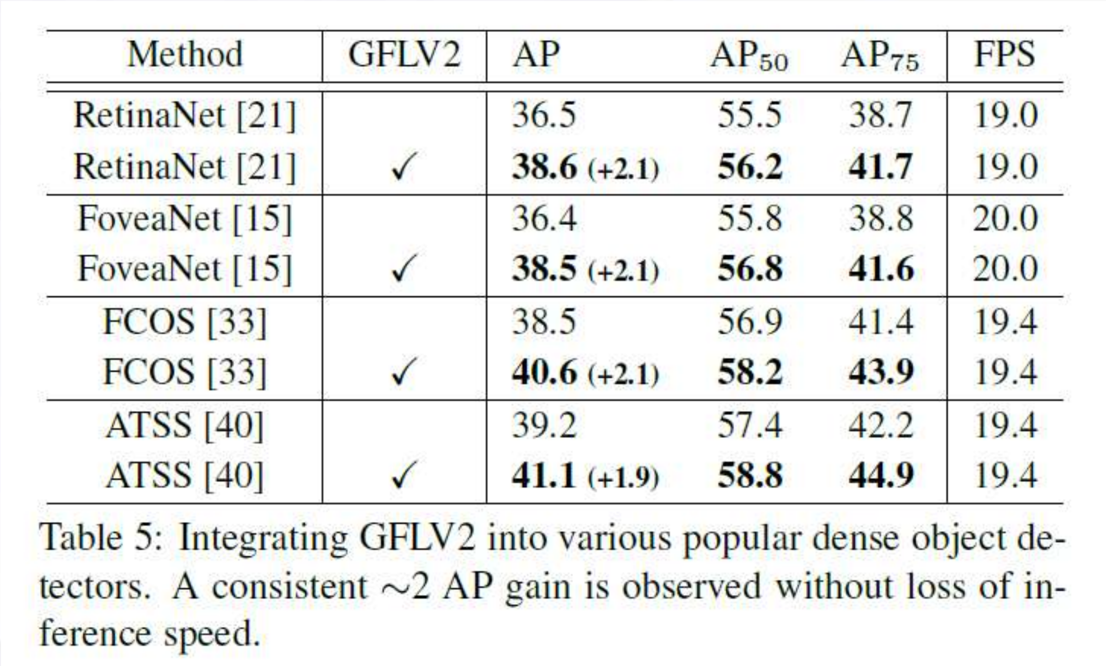
Table 3: Comparisons among different input features by fixing the hidden layer dimension of DGQP.

C
⊗
J
I
Decomposed Form
d
C
J
I
Composed Form

Figure 5: Different ways to utilize the distribution statistics, including Decomposed Form (left) and Composed Form (right).

5 Generalized Focal Loss V2

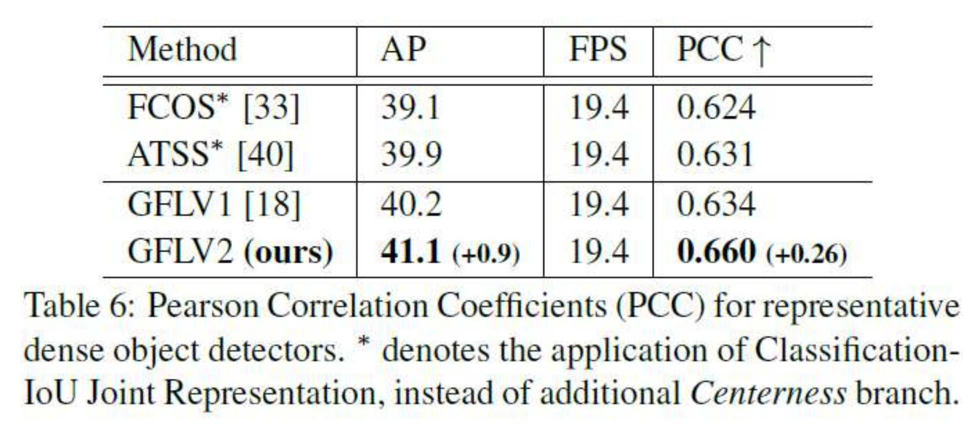
将GFLv2框架引入到其他的检测框架中



Method	GFLV2	AP	AP_{50}	AP_{75}	FPS
RetinaNet [21]		36.5	55.5	38.7	19.0
RetinaNet [21]	✓	**38.6** (+2.1)	**56.2**	**41.7**	19.0
FoveaNet [15]		36.4	55.8	38.8	20.0
FoveaNet [15]	✓	**38.5** (+2.1)	**56.8**	**41.6**	20.0
FCOS [33]		38.5	56.9	41.4	19.4
FCOS [33]	✓	**40.6** (+2.1)	**58.2**	**43.9**	19.4
ATSS [40]		39.2	57.4	42.2	19.4
ATSS [40]	✓	**41.1** (+1.9)	**58.8**	**44.9**	19.4

Table 5: Integrating GFLV2 into various popular dense object detectors. A consistent ∼2 AP gain is observed without loss of inference speed.

分别计算预测的IoU和真实IoU之间的(PCC)皮尔森相关系数



Method	AP	FPS	PCC ↑
FCOS* [33]	39.1	19.4	0.624
ATSS* [40]	39.9	19.4	0.631
GFLV1 [18]	40.2	19.4	0.634
GFLV2 (**ours**)	**41.1** (+0.9)	19.4	**0.660** (+0.26)

Table 6: Pearson Correlation Coefficients (PCC) for representative dense object detectors. * denotes the application of Classification-IoU Joint Representation, instead of additional *Centerness* branch.

5 Generalized Focal Loss V2

从训练损失上来看，GFL V2的DGQP成功地加速了训练过程并收敛到更低的损失

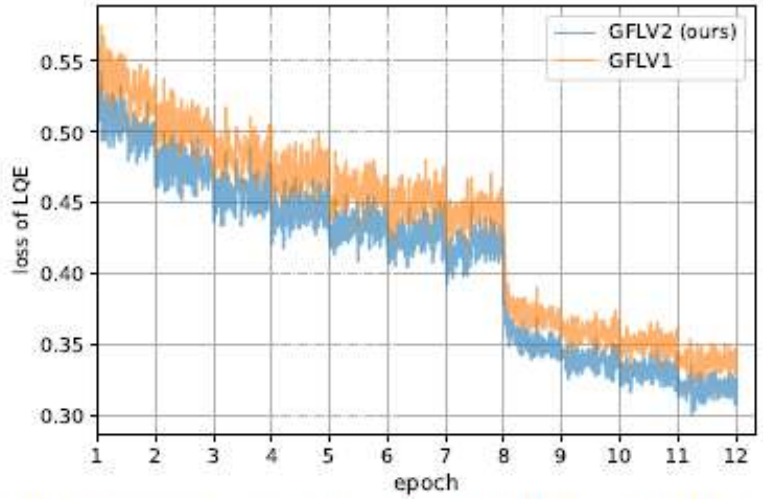
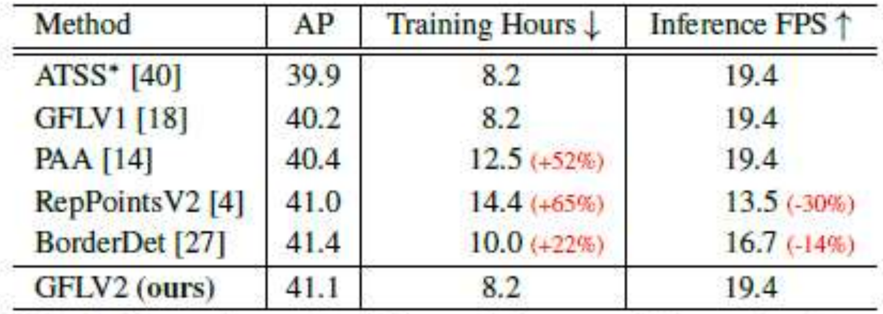


Figure 8: Comparisons of losses on LQE between GFLV1 and GFLV2. DGQP helps to ease the learning difficulty with lower losses during training.

训练效率和inference效率



Method	AP	Training Hours ↓	Inference FPS ↑
ATSS* [40]	39.9	8.2	19.4
GFLV1 [18]	40.2	8.2	19.4
PAA [14]	40.4	12.5 (+52%)	19.4
RepPointsV2 [4]	41.0	14.4 (+65%)	13.5 (-30%)
BorderDet [27]	41.4	10.0 (+22%)	16.7 (-14%)
GFLV2 (ours)	41.1	8.2	19.4

Table 8: Comparisons of training and inference efficiency based on ResNet-50 backbone. "Training Hours" is evaluated on 8 GeForce RTX 2080Ti GPUs under standard 1x schedule (12 epochs). * denotes the application of Classification-IoU Joint Representation.

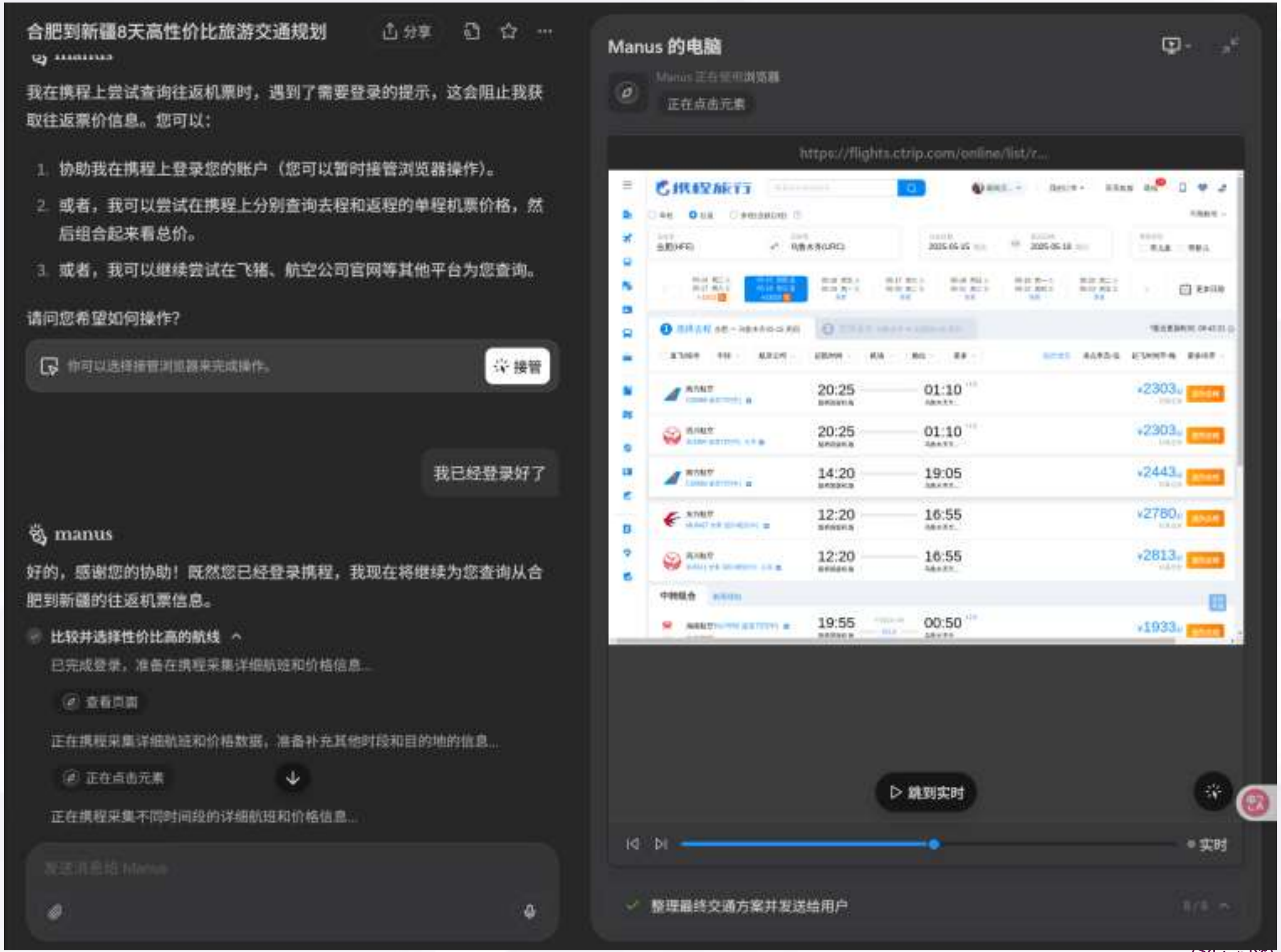
6

青稞

PC-Agent

6 PC-Agent

PC-Agent 与传统 LLM Agent 的区别		
推理流程	一次生成	多轮计划+反馈+控制
错误处理	基本没有	可重试、修改路径
可解释性	prompt黑箱	明确规划步骤和状态
灵活性	只能处理简单问题	可应对多工具、复杂场景
核心优势	模型本体强大	系统调度+结构引导



合肥到新疆8天高性价比旅游交通规划

我在携程上尝试查询往返机票时，遇到了需要登录的提示，这会阻止我获取往返票价信息。您可以：

1. 协助我在携程上登录您的账户（您可以暂时接管浏览器操作）。
2. 或者，我可以尝试在携程上分别查询去程和返程的单程机票价格，然后组合起来看总价。
3. 或者，我可以继续尝试在飞猪、航空公司官网等其他平台为您查询。

请问您希望如何操作?

你可以选择接管浏览器来完成操作。

我已经登录好了

manus

好的，感谢您的协助！既然您已经登录携程，我现在将继续为您查询从合肥到新疆的往返机票信息。

比较并选择性价比高的航线

已完成登录，准备在携程采集详细航班和价格信息...

正在携程采集详细航班和价格数据，准备补充其他时段和目的地的信息...

正在携程采集不同时间段的详细航班和价格信息...

Manus 的电脑

Manus 正在使用浏览器

https://flights.ctrip.com/online/list/r...

整理最终交通方案并发送给用户

7

技术脉络和个人思考

密集物体检测

7 技术演进趋势

趋势一：从样本数量关注 → 样本质量关注

Focal Loss	作者想到问题的出发点是密集目标检测中正负样本数量不均衡，关注的是样本数量
GFL-V1/V2	不再只关注“这是不是目标”，而关注“这个框质量到底好不好”——引入 IoU 指导的质量预测

趋势二：从框 → 无框 → 概率分布

传统 two-stage	大量 Anchor 框，回归偏移量
FCOS	摒弃 Anchor，直接回归四边距
GFL-V2	概率建模，模型有自己的不确定性和信心程度

趋势三：从一维回归 → 多维结构分布建模

FCOS	回归四个值：left/right/top/bottom
GFL-V2	对每个边距不是预测一个值，而是预测概率分布，再提取 top-k + mean 特征，用于 IoU 打分

7 个人思考

思考一：Focal Loss中γ参数是否太依赖经验性参数?

在原文中，**γ=2被写死为最佳参数**，但这是基于COCO数据集，并不会根据不同的数据集和图形**自适应地调整自己的参数**。

个人想法：是否可以在训练过程中让γ 随着loss进行动态的变化，然后取最优结果？

思考二：分布预测这一阶段可能计算消耗会更多?

例如FCOS预测的只是4个方向上的参数，而GFL-V2要预测四个方向上的分布向量，还要进行Top-k，mean等一系列计算明显会比FCOS计算消耗更多

个人想法：是否可以先进行传统分数估计，然后再对得分高的目标框使用DGQP？

思考三：未来的目标检测，是否不止于“框”？

现在这个框只是简单的识别目标的位置，那么是否也可以增添一些其他的方向，比如这个目标下一步**移动的趋势**？多个目标之间**是否有关系**，比如识别出的人物是在对话还是在交谈？

THANKS

感谢聆听

- 院系：人工智能学院
- 专业：工科试验班
- 答辩人：黄子豪