Deformable DETR: 基于稀疏空间采样的注意力机制，让DCN与Transformer一起玩！ - 知乎 (zhihu.com)

Abstract

DETR最近被提出，旨在消除目标检测中许多手工设计的组件的需求，并展示出良好的性能。然而，由于Transformer注意力模块在处理图像特征图时的限制，DETR存在收敛速度慢和特征空间分辨率有限的问题。为了缓解这些问题，我们提出了可变形DETR，其**注意力模块仅关注参考点周围的一小组重要采样点**。可变形DETR可以在比DETR少10倍的训练迭代次数下取得更好的性能（尤其是对小物体）。在COCO基准测试上进行了大量实验证明了我们方法的有效性。代码已在https://github.com/fundamentalvision/Deformable-DETR 上发布。

1 INTRODUCTION

现代物体检测器采用许多手工制作的组件（Liu等人，2020），例如锚点生成、基于规则的训练目标分配、非最大值抑制（NMS）后处理。它们不是完全端到端的。最近，Carion等人（2020）提出了DETR，以消除对这些手工制作组件的需求，并构建了第一个完全端到端的物体检测器，实现了非常有竞争力的性能。DETR利用了简单的体系结构，通过结合卷积神经网络（CNNs）和Transformer（Vaswani等人，2017）编码器-解码器。它们利用Transformer的多才多艺和强大的关系建模能力来替代手工制作的规则，在经过适当设计的训练信号下实现。

尽管DETR具有有趣的设计和良好的性能，但它也有自己的问题：(1)与现有的目标检测器相比，DETR需要更长的训练周期才能收敛。例如，在COCO（Lin等人，2014）基准上，DETR需要500个训练周期才能收敛，这比Faster R-CNN（Ren等人，2015）慢10到20倍左右。(2)DETR在检测小物体时的性能相对较低。现代的目标检测器通常利用多尺度特征，从小物体检测到高分辨率特征图。与此同时，高分辨率特征图导致DETR的复杂性不可接受。上述问题主要由处理图像特征图时Transformer组件的不足所致。在初始化时，注意力模块对特征图中的所有像素几乎施加均匀的注意力权重。需要很长的训练周期才能使注意力权重聚焦在稀疏有意义的位置上。另一方面，Transformer编码器中的注意力权重计算相对于像素数的计算量是二次方的。因此，处理高分辨率特征图的计算和内存复杂性非常高。

在图像域中，可变形卷积(DCN)（Dai等人，2017）是一种强大而有效的机制，可以关注稀疏空间位置。它自然避免了上述问题。虽然它缺乏元素关系建模机制，而这是DETR成功的关键。

在本文中，我们提出了Deformable DETR，它缓解了DETR收敛慢和高复杂性的问题。它结

合了可变形卷积的稀疏空间采样和Transformer的关系建模能力。我们提出了可变形注意力模块，它关注于特征图像素中的一小部分采样位置作为突出关键元素的预滤波器。该模块可以自然地扩展到聚合多尺度特征，而无需使用FPN（Lin等人，2017a）。在Deformable DETR中，我们利用（多尺度）可变形注意力模块来替换处理特征图的Transformer注意力模块，如图1所示。

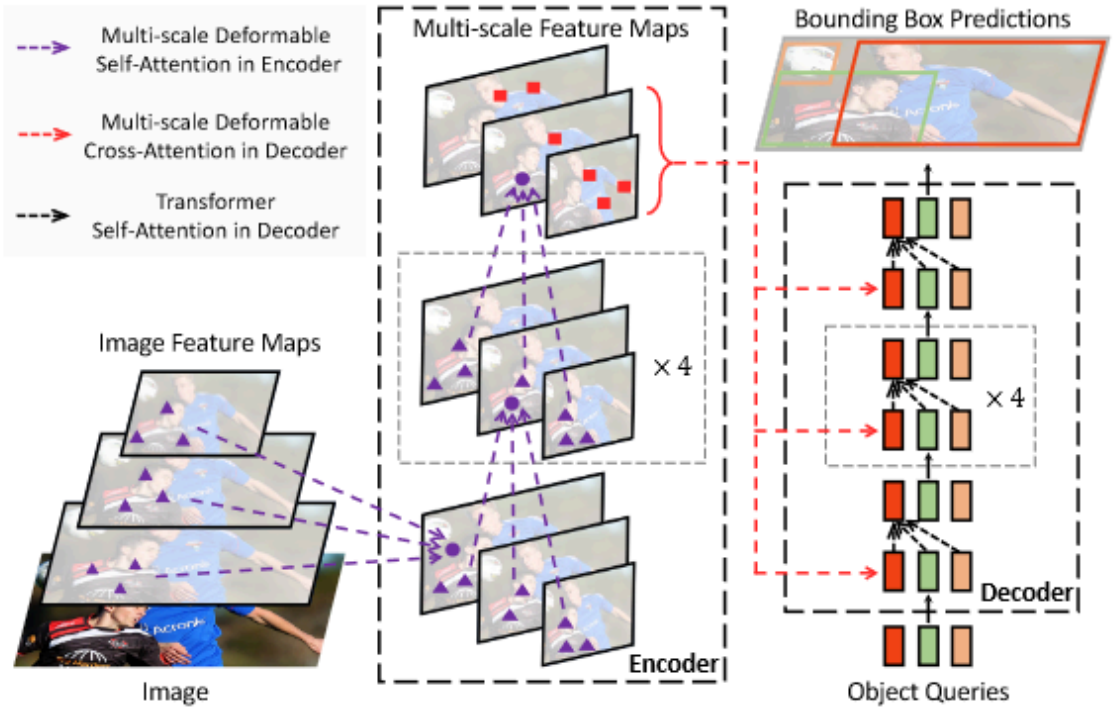

Figure 1: Illustration of the proposed Deformable DETR object detector.

Deformable DETR由于其快速收敛以及计算和内存的高效性，为我们提供了开发各种端到端目标检测器的可能性。我们探索了一种简单而有效的迭代边界框细化机制来提高检测性能。我们还尝试了一个两阶段可变形DETR，其中区域提议也是由一种变体的可变形DETR生成的，这些提议被进一步馈送到解码器进行迭代边界框细化。

在COCO（Lin等人，2014）基准的大量实验证明了我们的方法的有效性。与DETR相比，Deformable DETR可以在少10倍的训练周期下实现更好的性能（特别是在小物体上）。提出的两阶段可变形DETR的变体可以进一步提高性能。代码发布在https://github.com/fundamentalvision/Deformable-DETR 上。

2 RELATED WORK

高效的注意力机制

Transformer（Vaswani等人，2017）涉及自注意力和交叉注意力机制。在Transformer中最广为人知的一个问题是，在大量键元素的情况下，时间和内存复杂度很高，这阻碍了模型的可扩展性在很多情况下。最近，已经做出了许多努力来解决这个问题（Tay等人，2020b），这些努力在实践中大致可以分为三类。

第一类是使用预先定义的稀疏注意力模式。最直接的方法是限制注意力模式为固定的局部窗口。大多数工作（Liu等人，2018a；Parmar等人，2018；Child等人，2019；Huang等人，2019；Ho等人，2019；Wang等人，2020a；Hu等人，2019；Ramachandran等人，2019；Qiu等人，2019；Beltagy等人，2020；Ainslie等人，2020；Zaheer等人，2020）遵循这一范式。虽然将注意力模式限制在局部邻域可以降低复杂性，但会失去全局信息。为了补偿，Child等人（2019）；Huang等人（2019）；Ho等人（2019）；Wang等人（2020a）以固定间隔对关键元素进行关注，以显着增加键的感受野。Beltagy等人（2020）；Ainslie等人（2020）；Zaheer等人（2020）允许少数特殊令牌访问所有键元素。Zaheer等人（2020）；Qiu等人（2019）还添加了一些预先固定的稀疏注意力模式来直接关注远距离的关键元素。

第二类是学习数据依赖的稀疏注意力。Kitaev等人（2020）提出了局部敏感哈希（LSH）的注意力，将查询和键元素哈希到不同的bin。Roy等人（2020）提出了一个相似的想法，k-means找到了最相关的键。Tay等人（2020a）学习了块置换以进行块状稀疏注意力。

第三类是探索自注意力中的低秩性质。Wang等人（2020b）通过在大小维度而不是通道维度上进行线性投影来减少键元素数量。Katharopoulos等人（2020）；Choromanski等人（2020）通过核化近似重写了自注意力的计算。

在图像领域，高效注意力机制的设计（例如Parmar等人（2018）；Child等人（2019）；Huang等人（2019）；Ho等人（2019）；Wang等人（2020a）；Hu等人（2019）；Ramachandran等人（2019））仍然局限于第一类。尽管在理论上降低了复杂性，但Ramachandran等人（2019）；Hu等人（2019）承认，与具有相同FLOPs的传统卷积相比，这种方法的实现速度慢得多（至少慢3倍），这是由于内存访问模式的固有局限性。

另一方面，正如Zhu等人（2019a）所讨论的那样，存在变体的卷积，例如可变形卷积（Dai等人，2017；Zhu等人，2019b）和动态卷积（Wu等人，2019），这些也可以视为自注意力机制。特别是，可变形卷积在图像识别上的操作比Transformer自注意力更有效和更高效。同时，它缺乏元素关系建模机制。

我们提出的可变形注意力模块受到可变形卷积的启发，属于第二类。它只关注由查询元素特征预测的一小固定集合的采样点。与Ramachandran等人（2019）；Hu等人（2019）不同，可变形注意力在相同FLOPs下仅比传统卷积慢一点。

用于对象检测的多尺度特征表示

目标检测的主要难点之一是有效地表示尺度差异很大的目标。现代目标检测器通常利用多尺度特征来适应这一点。作为开创性工作之一，FPN（Lin等人，2017a）提出了一种自上而下的路径来组合多尺度特征。PANet（Liu等人，2018b）在FPN的基础上进一步增加了一条自下而上的路径。Kong等人（2018）通过全局注意力操作组合所有尺度的特征。Zhao等人（2019）提出了一种U形模块来融合多尺度特征。最近，NAS-FPN（Ghiasi等人，2019）和Auto-FPN（Xu等人，2019）通过神经架构搜索自动设计跨尺度连接。Tan等人（2020）提出了BiFPN，它是PANet的简化版重复。我们提出的多尺度可变形注意力模块可以通过注意力机制自然地聚合多尺度特征图，无需这些特征金字塔网络的帮助。

3 REVISITING TRANSFORMERS AND DETR

在Transformer模型中使用了多头注意力机制

Transformer模型是基于注意力机制的网络架构，用于机器翻译任务。给定一个查询元素（例如，输出句子中的目标词）和一组键元素（例如，输入句子中的源词），多头注意力模块根据衡量查询-键对兼容性的注意力权重自适应地聚合键内容。为了允许模型关注来自不同表示子空间和不同位置的内容，不同注意力头的输出通过可学习的权重进行线性聚合。其中$q \in \Omega_q$具有表示特征$z_q \in \mathbb{R}^C$的查询元素，$k \in \Omega_k$具有表示特征$x_k \in \mathbb{R}^C$的键元素，其中C是特征维度，Ω_q和Ω_k分别指定查询元素和键元素的集合。然后，多头注意力特征通过以下方式计算：

$$MultiHeadAttn(z_q, x) = \sum_{m=1}^{M} W_m[\sum_{k \in \Omega_k} A_{mqk} \cdot W'_m x_k] \tag{1}$$

这里m索引注意力头，$W'_m \in \mathbb{R}^{C_v \times C}$和$W_m \in \mathbb{R}^{C \times C_v}$都是可学习的权重（默认情况下$C_v = C/M$）。注意力权重$A_{mqk} \propto \exp \frac{z_q^T U_m^T V_m x_k}{\sqrt{C_v}}$，并归一化，即$\sum_k \in \Omega_k A_{mak} = 1$，其中$U_m, V_m \in \mathbb{R}^{C_v \times C}$也是可学习的权重。为了消除不同空间位置的歧义，表示特征z_q和x_k通常是元素内容和位置嵌入的连接/求和。

Transformer有两个已知的问题。一个是Transformer需要很长的训练计划才能收敛。假设查询元素和键元素的数量分别为Nq和Nk。通常情况下，如果参数初始化得当，$U_m z_q$和$V_m x_k$遵循均值为0、方差为1的分布，这使得注意力权重$A_{mqk} \approx \frac{1}{N_k}$，当Nk很大时。这会导致输入特征的梯度模糊。因此，需要很长的训练计划，以便注意力权重能够专注于特定的键。在图像域中，键元素通常是图像像素，Nk可能非常大，导致收敛过程变得繁琐。

另一方面，对于多头注意力来说，具有大量查询元素和键元素的计算和内存复杂度可能非常

高。等式1的计算复杂度为$O(N_qC^2 + N_kC^2 + N_qN_kC)$。在图像域中，查询元素和键元素都是像素，$N_q = N_K \gg C$，复杂度由第三项主导，即$O(N_qN_kC)$。因此，多头注意力模块随着特征图尺寸的增加而遭受二次复杂度增长的影响。

DETR

DETR（Carion等人，2020）建立在Transformer编码器-解码器架构之上，并结合了基于集合的匈牙利损失，该损失通过二部匹配强制每个真实边界框的唯一预测。我们简要回顾了网络架构，如下所示。

DETR利用标准Transformer编码器-解码器架构将CNN主干（例如ResNet（He等人，2016））提取的输入特征图$x \in \mathbb{R}^{C\times H\times W}$转换为一系列目标查询的特征。在目标查询特征（由解码器产生）的顶部添加了一个3层前馈神经网络（FFN）和一个线性投影，作为检测头。FFN作为回归分支来预测边界框坐标$b \in [0, 1]^4$，其中b = {bx, by, bw, bh}编码了归一化的框中心坐标、框高度和宽度（相对于图像大小）。线性投影作为分类分支来产生分类结果。

对于DETR中的Transformer编码器，查询和键元素都是特征图中的像素。输入是带有编码位置嵌入的ResNet特征图。让H和W分别表示特征图的高度和宽度。自注意力的计算复杂性为$O(H^2W^2C)$，随着空间尺寸的增加而呈二次方增长。

DETR中的Transformer解码器的输入包括来自编码器的特征图和由可学习位置嵌入表示的N个对象查询（例如，N = 100）。解码器中有两种注意力模块，即交叉注意力模块和自注意力模块。在交叉注意力模块中，对象查询从特征图中提取特征。查询元素是对象查询，键元素是编码器输出的特征图。在这种情况下，$N_q = N$，$N_k = H \times W$，交叉注意力的复杂性为$O(HWC^2 + NHWC)$，它随着特征图空间尺寸的增加而线性增长。在自注意力模块中，对象查询相互交互，以捕捉它们之间的关系。查询和键元素都是对象查询。在这种情况下，$N_q = N_k = N$，自注意力模块的复杂性为$O(2NC^2 + N^2C)$，对于适度数量的对象查询来说，这是可以接受的。

DETR是一个很有吸引力的目标检测设计，它消除了对许多手工设计的组件的需求。然而，它也有自己的问题。这些问题主要归因于处理图像特征图作为关键元素的Transformer注意力的缺陷：(1) DETR在检测小物体方面性能相对较低。现代目标检测器使用高分辨率特征图来更好地检测小物体。然而，高分辨率特征图会导致DETR中Transformer编码器的自注意力模块的复杂性不可接受，该模块的复杂性随着输入特征图的空间大小呈二次方增长。(2)与现代目标检测器相比，DETR需要更多的训练轮次才能收敛。这主要是因为处理图像特征的注意力模块很难训练。例如，在初始化时，交叉注意力模块几乎对整个特征图进行平均注意力。然而，在训练结束时，注意力图被学习为非常稀疏，只关注对象的极端。似乎DETR需要很长的训练计划来学习注意力图中的这种显著变化。

4 METHOD

4.1 DEFORMABLE TRANSFORMERS FOR END-TO-END OBJECT DETECTION

可变形注意力模块

将Transformer注意力应用于图像特征图的核心问题是它会查看所有可能的空间位置。为了解决这个问题，我们提出了一个可变形的注意力模块。受到可变形卷积（Dai等人，2017年；Zhu等人，2019b）的启发，可变形的注意力模块只关注围绕参考点的少量关键采样点，而不考虑特征图的空间大小，如图2所示。通过为每个查询分配一个小的固定数量的键，可以减轻收敛问题和特征空间分辨率问题。

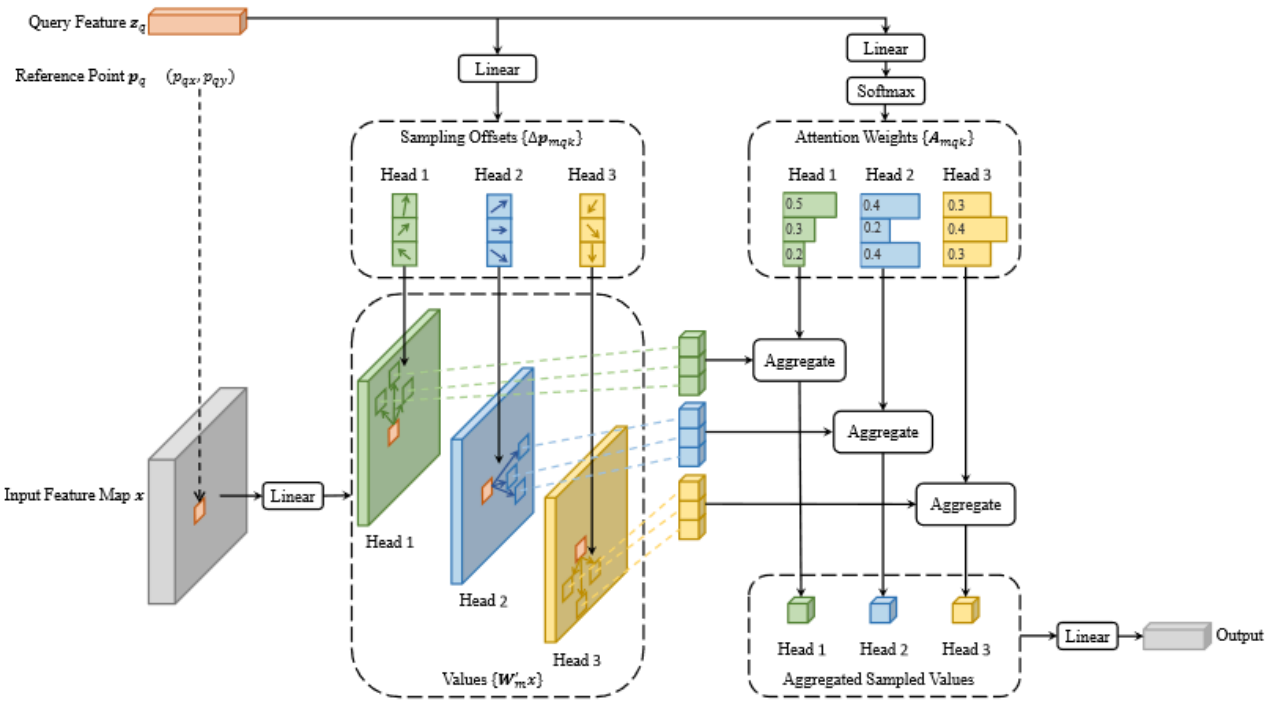

Figure 2: Illustration of the proposed deformable attention module.

对于输入特征图 $x \in \mathbb{R}^{C \times H \times W}$，使用索引 q 来查询一个具有内容特征 z_q 和 2-d 参考点 p_q 的元素，可变形注意力特征通过以下方式计算：

$$DeformAttn(z_q, p_q, x) = \sum_{m=1}^{M} W_m [\sum_{k=1}^{K} A_{mqk} \cdot W'_m x(p_q + \Delta p_{mqk})] \qquad (2)$$

其中m是注意力头的索引，k是采样的键的索引，K是总采样键数（$K \ll HW$）。Δp_{mqk}和

A_{mqk}分别表示第m个注意力头中第k个采样点的采样偏移和注意力权重。标量注意力权重A_{mqk}在范围[0,1]内，通过归一化$\sum_{k=1}^{K} A_{mqk} = 1$。$\Delta p_{mqk} \in \mathbb{R}^2$是二维实数，取值范围不受限制。由于$p_q + \Delta p_{mqk}$是分数，因此在计算$x(p_q + \Delta p_{mqk})$时采用双线性插值，类似于Dai等人（2017年）的方法。Δp_{mqk}和A_{mqk}都通过线性投影得到，投影操作应用于查询特征z_q。在实现中，查询特征z_q被馈送到具有$3MK$通道的线性投影运算符，其中前$2MK$通道编码采样偏移Δp_{mqk}，剩余的MK通道被馈送到softmax运算符以获得注意力权重A_{mqk}。

这个变形注意力模块是专门用于处理卷积特征图的。定义查询元素的数量为N_q，在MK相对较小的情况下，变形注意力模块的复杂度为$O(2N_qC^2 + \min(HWC^2, N_qKC^2))$（详见附录A.1）。当应用于DETR编码器时，其中$N_q = HW$，复杂度变为$O(HWC^2)$，与空间尺寸的线性复杂度成正比。而当应用为DETR解码器中的交叉注意力模块时，其中$N_q = N$（N为对象查询的数量），复杂度变为$O(NKC^2)$，与空间尺寸HW无关。

多尺度变形注意力模块

大多数现代目标检测框架都受益于多尺度特征图（Liu等，2020）。我们提出的变形注意力模块可以自然地扩展到多尺度特征图。

设 $\{x^l\}_{l=1}^{L}$ 为输入的多尺度特征图，其中$x^l \in \mathbb{R}^{C \times H_l \times W_l}$。设$\hat{p}_q \in [0,1]^2$ 为每个查询元素 q 的归一化坐标，那么多尺度变形注意力模块的应用形式为：

$$\begin{aligned} &MSDeformAttn(z_q, \hat{p}_q, \{x^l\}_{l=1}^{L}) \\ &= \sum_{m=1}^{M} W_m [\sum_{l=1}^{L} \sum_{k=1}^{K} A_{mlqk} \cdot W'_m x^l(\phi_l(\hat{p}_q) + \Delta p_{mlqk})] \end{aligned} \tag{3}$$

其中，m 索引注意头，l 索引输入特征级别，k 索引采样点。Δp_{mlqk} 和 A_{mlqk} 分别表示第 l 特征级别和第 m 注意头中第 k 个采样点的采样偏移和注意力权重。标量注意力权重 A_{mlqk} 通过 $\sum_{l=1}^{L} \sum_{k=1}^{K} A_{mlqk} = 1$ 进行归一化。在这里，我们使用归一化坐标 $\hat{p}_q \in [0,1]^2$ 以清晰表示尺度公式，其中归一化坐标 (0, 0) 和 (1, 1) 分别表示图像的左上角和右下角。方程 3 中的函数$\phi_l(\hat{p}_q)$ 将归一化坐标$\hat{p}_q$ 重新缩放到第 l 特征级别的输入特征图。多尺度变形注意力与之前的单尺度版本非常相似，只是从多尺度特征图中采样LK个点，而不是从单尺度特征图中采样 K 个点。

所提出的注意力模块在 $L = 1$、$K = 1$ 且 $W'_m \in \mathbb{R}^{C_v \times C}$ 固定为单位矩阵时将退化为可变形卷积（Dai 等人，2017）。可变形卷积设计用于单尺度输入，每个注意力头仅关注一个采样点。然而，我们的多尺度可变形注意力从多尺度输入中查看多个采样点。所提出的（多尺度）可变形注意力模块也可以看作是 Transformer 注意力的一种高效变体，其中通过可变形采样位置引入了一种预过滤机制。当采样点遍历所有可能的位置时，所提出的注意力模块等效于 Transformer 注意力。

可变形 Transformer 编码器。我们用所提出的多尺度可变形注意力模块替换 DETR 中处理特征图的 Transformer 注意力模块。编码器的输入和输出都是相同分辨率的多尺度特征图。在编码器中，我们从 ResNet（He 等人，2016）的 C3 到 C5 阶段的输出特征图中提取多尺度特征图$\{x^l\}_{l=1}^{L-1}(L=4)$，通过一个 1 × 1 卷积进行转换，其中 C_l 的分辨率比输入图像低2^l。最低分辨率的特征图 x^L 通过对最终 C_5 阶段进行 3 × 3 步幅 2 卷积获得，表示为C_6。所有多尺度特征图都具有 C = 256 个通道。请注意，不使用 FPN（Lin 等人，2017a）中的自顶向下结构，因为我们提出的多尺度可变形注意力本身就可以在多尺度特征图之间交换信息。构建多尺度特征图的过程也在附录 A.2 中说明。在第 5.2 节的实验证明，添加 FPN 不会提高性能。

在对编码器中的多尺度可变形注意力模块的应用中，输出是与输入相同分辨率的多尺度特征图。键和查询元素都是来自多尺度特征图的像素。对于每个查询像素，参考点是它自己。为了确定每个查询像素位于哪个特征级别，我们添加了一个尺度级别嵌入，表示为el，到特征表示中，除了位置嵌入之外。与具有固定编码的位置嵌入不同，尺度级别嵌入 $\{e_l\}_{l=1}^{L}$ 是随机初始化的，并与网络一起进行联合训练。

可变形Transformer解码器

解码器中有交叉注意力和自注意力模块。两种类型的注意力模块的查询元素都是对象查询。在交叉注意力模块中，对象查询从特征图中提取特征，其中键元素来自编码器的输出特征图。在自注意力模块中，对象查询彼此交互，其中键元素是对象查询。由于我们提出的可变形注意力模块是为了处理卷积特征图而设计的，我们只将每个交叉注意力模块替换为多尺度可变形注意力模块，而保持自注意力模块不变。对于每个对象查询，通过可学习的线性投影和sigmoid函数，从其对象查询嵌入中预测参考点的二维归一化坐标 $\hat{p}_q$。

由于多尺度可变形注意力模块提取了参考点周围的图像特征，我们让检测头相对于参考点预测边界框的相对偏移，以进一步降低优化难度。参考点被用作框中心的初始猜测。检测头相对于参考点预测相对偏移。有关详细信息，请参见附录A.3。通过这种方式，学习到的解码器注意力将与预测的边界框具有强相关性，这也加速了训练的收敛。

通过在DETR中用可变形注意力模块替换Transformer注意力模块，我们建立了一种高效且收敛速度快的检测系统，称为可变形DETR（见图1）。

4.2 ADDITIONAL IMPROVEMENTS AND VARIANTS FOR DEFORMABLE DETR

可变形DETR为我们利用各种改进和变体的端到端目标检测器提供了可能性，由于篇幅有限，我们在这里只介绍了这些改进和变体的核心思想。实现细节请参见附录A.4。

迭代边界框细化

这受到光流估计中开发的迭代细化的启发（Teed＆Deng，2020）。我们建立了一个简单而有效的迭代边界框细化机制，以提高检测性能。在这里，每个解码器层根据前一层的预测对边界框进行细化。

两阶段Deformable DETR

在原始的DETR中，解码器中的对象查询与当前图像无关。受到两阶段目标检测器的启发，我们探索了Deformable DETR的一个变体，用于生成区域提案作为第一阶段。生成的区域提案将被馈送到解码器作为对象查询进行进一步的细化，形成两阶段的Deformable DETR。

在第一阶段，为了实现高召回提案，多尺度特征图中的每个像素都将用作对象查询。然而，直接将对象查询设置为像素将带来解码器中自注意模块的无法接受的计算和内存成本，其复杂度与查询数量的平方成正比。为了避免这个问题，我们移除解码器，形成一个仅包含编码器的Deformable DETR，用于生成区域提案。在其中，每个像素被分配为一个对象查询，它直接预测一个边界框。选择分数最高的边界框作为区域提案。在将区域提案馈送到第二阶段之前，不应用非极大值抑制。

5 EXPERIMENT

数据集

我们在COCO 2017数据集上进行实验（Lin等人，2014年）。我们的模型在训练集上进行训练，并在验证集和测试集上进行评估。

实现细节

我们在剖析时使用了ImageNet（Deng等人，2009年）预训练的ResNet-50（He等人，2016年）作为骨干网络。多尺度特征图是在没有FPN（Lin等人，2017a年）的情况下提取的。默认情况下，对于可变注意力，M = 8，K = 4。可变Transformer编码器的参数在不同特征级别之间共享。其他超参数设置和训练策略主要遵循DETR（Carion等人，2020年），除了使用Focal Loss（Lin等人，2017b年）进行边界框分类，其损失权重为2，并且将对象查询的数量从100增加到300。我们还报告了DETR-DC5在这些修改下的性能，以进行公正比较，标记为DETR-DC5+。默认情况下，模型经过50个epoch的训练，并且在第40个epoch时，学习率衰减为0.1倍。与DETR（Carion等人，2020年）一样，我们使用Adam优化器（Kingma＆Ba，2015年）进行模型训练，基本学习率为2×10^{-4}，β1 = 0.9，β2 = 0.999，权重衰减为10^{-4}。用于预测对象查询参考点和采样偏移的线性投影的学习率乘以0.1。运行时间是在NVIDIA Tesla V100 GPU上评估的。

5.1 COMPARISON WITH DETR

如表1所示，与Faster R-CNN + FPN相比，DETR需要更多的训练轮次才能收敛，并在检测小目标方面性能较差。与DETR相比，Deformable DETR在10倍更少的训练轮次内实现了更好的性能（尤其是在小目标上）。详细的收敛曲线见图3。在迭代边界框细化和两阶段范例的帮助下，我们的方法可以进一步提高检测精度。

我们提出的可变形DETR具有与Faster R-CNN+FPN和DETR-DC5相当的FLOP。但运行速度比DETR-DC5快得多（1.6倍），仅比faster R-CNN+FPN慢25%。DETR-DC5的速度问题主要是由于Transformer注意到大量的内存访问。我们提出的可变形注意力可以缓解这个问题，代价是无序的内存访问。因此，它仍然比传统的卷积稍微慢一些。

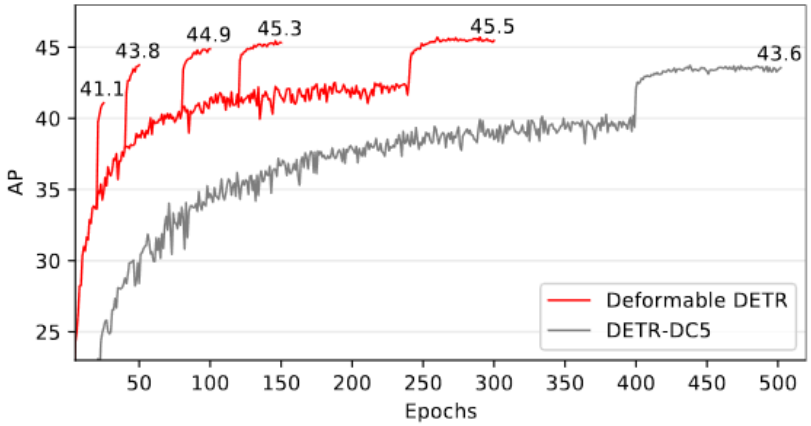

Figure 3: Convergence curves of Deformable DETR and DETR-DC5 on COCO 2017 val set. For Deformable DETR, we explore different training schedules by varying the epochs at which the learning rate is reduced (where the AP score leaps).

Table 1: Comparision of Deformable DETR with DETR on COCO 2017 val set. DETR-DC5$^+$ denotes DETR-DC5 with Focal Loss and 300 object queries.

Method	Epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	params	FLOPs	Training GPU hours	Inference FPS
Faster R-CNN + FPN	109	42.0	62.1	45.5	26.6	45.4	53.4	42M	180G	380	26
DETR	500	42.0	62.4	44.2	20.5	45.8	61.1	41M	86G	2000	28
DETR-DC5	500	43.3	63.1	45.9	22.5	47.3	61.1	41M	187G	7000	12
DETR-DC5	50	35.3	55.7	36.8	15.2	37.5	53.6	41M	187G	700	12
DETR-DC5$^+$	50	36.2	57.0	37.4	16.3	39.2	53.9	41M	187G	700	12
Deformable DETR	50	43.8	62.6	47.7	26.4	47.1	58.0	40M	173G	325	19
+ iterative bounding box refinement	50	45.4	64.7	49.0	26.8	48.3	61.7	40M	173G	325	19
++ two-stage Deformable DETR	50	46.2	65.2	50.0	28.8	49.2	61.7	40M	173G	340	19

5.2 ABLATION STUDY ON DEFORMABLE ATTENTION

表2展示了提出的可变形注意力模块的各种设计选择的消融结果。使用多尺度输入而不是单一

尺度输入可以有效提高检测准确性，AP提高了1.7%，尤其是在小目标上，APS提高了2.9%。增加采样点数K可以进一步提高0.9%的AP。使用多尺度可变形注意力，允许在不同尺度水平上进行信息交换，可以额外提高1.5%的AP。由于已经采用了跨级别的特征交换，添加FPN不会提高性能。当不应用多尺度注意力且K = 1时，我们的（多尺度）可变形注意力模块会退化为可变形卷积，准确性明显降低。

Table 2: Ablations for deformable attention on COCO 2017 val set. "MS inputs" indicates using multi-scale inputs. "MS attention" indicates using multi-scale deformable attention. K is the number of sampling points for each attention head on each feature level.

MS inputs	MS attention	K	FPNs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
✓	✓	4	FPN (Lin et al., 2017a)	43.8	62.6	47.8	26.5	47.3	58.1
✓	✓	4	BiFPN (Tan et al., 2020)	43.9	62.5	47.7	25.6	47.4	57.7
		1	w/o	39.7	60.1	42.4	21.2	44.3	56.0
✓		1		41.4	60.9	44.9	24.1	44.6	56.1
✓		4		42.3	61.4	46.0	24.8	45.1	56.3
✓	✓	4		43.8	62.6	47.7	26.4	47.1	58.0

5.3 COMPARISON WITH STATE-OF-THE-ART METHODS

表格3比较了所提出的方法与其他最先进的方法。我们的模型在表格3中同时使用了迭代边界框优化和两阶段机制。使用ResNet-101和ResNeXt-101（Xie等人，2017）时，我们的方法无需额外装饰即可分别实现48.7 AP和49.0 AP的精度。当使用带有DCN（Zhu等人，2019b）的ResNeXt-101时，精度提升至50.1 AP。在附加测试时间增强时，所提出的方法实现了52.3 AP的精度。

Table 3: Comparison of Deformable DETR with state-of-the-art methods on COCO 2017 test-dev set. "TTA" indicates test-time augmentations including horizontal flip and multi-scale testing.

Method	Backbone	TTA	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
FCOS (Tian et al., 2019)	ResNeXt-101		44.7	64.1	48.4	27.6	47.5	55.6
ATSS (Zhang et al., 2020)	ResNeXt-101 + DCN	✓	50.7	68.9	56.3	33.2	52.9	62.4
TSD (Song et al., 2020)	SENet154 + DCN	✓	51.2	71.9	56.0	33.8	54.8	64.2
EfficientDet-D7 (Tan et al., 2020)	EfficientNet-B6		52.2	71.4	56.3	-	-	-
Deformable DETR	ResNet-50		46.9	66.4	50.8	27.7	49.7	59.9
Deformable DETR	ResNet-101		48.7	68.1	52.9	29.1	51.5	62.0
Deformable DETR	ResNeXt-101		49.0	68.5	53.2	29.7	51.7	62.8
Deformable DETR	ResNeXt-101 + DCN		50.1	69.7	54.6	30.6	52.8	64.7
Deformable DETR	ResNeXt-101 + DCN	✓	52.3	71.9	58.1	34.4	54.4	65.6

6 CONCLUSION

可变形DETR是一种端到端的目标检测器，它高效且收敛速度快。它使我们能够探索更有趣和实用的端到端目标检测器的变体。可变形DETR的核心是（多尺度）可变形注意力模块，这是一种在处理图像特征图时的有效的注意力机制。我们希望我们的工作为探索端到端目标检测开启了新的可能性。

A APPENDIX

A.1 COMPLEXITY FOR DEFORMABLE ATTENTION

假设查询元素的数量为Nq，在可变注意力模块（参见方程2）中，计算采样坐标偏移Δp_{mqk}和注意力权重A_{mqk}的复杂度为$O(3N_qCMK)$。给定采样坐标偏移和注意力权重，计算方程2的复杂度为$O(N_qC^2 + N_qKC^2 + 5N_qKC)$，其中$5N_qKC$中的5是由于双线性插值和注意力中的加权和。另一方面，我们也可以在采样之前计算$W'_m x$，因为它与查询无关，计算方程2的复杂度将变为$O(N_qC^2 + HWC^2 + 5N_qKC)$。因此，可变注意力的总复杂度为$O(N_qC^2 + min(HWC^2, N_qKC^2) + 5N_qKC + 3N_qCMK)$。在我们的实验中，M = 8，K ≤ 4，C = 256，默认情况下5K + 3M K < C，复杂度为$O(2N_qC^2 + min(HWC^2, N_qKC^2))$。

A.2 CONSTRUCTING MULT-SCALE FEATURE MAPS FOR DEFORMABLE DETR

如第4.1节所述和图4所示，编码器的输入多尺度特征图$\{x^l\}_{l=1}^{L-1}(L = 4)$是从ResNet中$C_3$至$C_5$级的输出特征图中提取的（He et al.，2016）（通过1×1卷积变换）。在最后的C5阶段，通过3×3步长2卷积获得最低分辨率的特征图x^L。注意，没有使用FPN（Lin et al.，2017a），因为我们提出的多尺度可变形注意力本身可以在多尺度特征图之间交换信息。

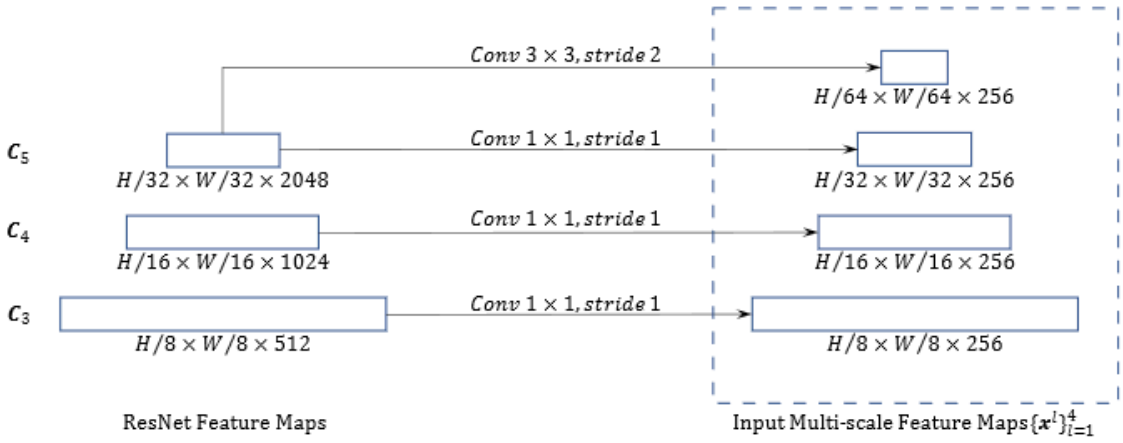

Figure 4: Constructing mult-scale feature maps for Deformable DETR.

A.3 BOUNDING BOX PREDICTION IN DEFORMABLE DETR

由于多尺度可变形注意力模块提取了围绕参考点的图像特征，我们设计了检测头来预测边界框，使其成为相对于参考点的相对偏移，以进一步减少优化难度。参考点被用作边界框中心的初始猜测。检测头预测相对于参考点的相对偏移$\hat{p}_q = (\hat{p}_{qx}, \hat{p}_{qy})$，即$\hat{b}_q = \{\sigma(b_{qx} + \sigma^{-1}(\hat{p}_{qx})), \sigma(b_{qy} + \sigma^{-1}(\hat{p}_{qy})), \sigma(b_{qw}), \sigma(b_{qh})\}$，其中$b_{q\{x,y,w,h\}} \in \mathbb{R}$由检测头预测。σ和$\sigma^{-1}$分别表示sigmoid和逆sigmoid函数。使用σ和σ^{-1}是为了确保$\hat{b}$是规范化坐标，即$\hat{b}_q \in [0, 1]^4$。通过这种方式，学到的解码器注意力将与预测的边界框具有强烈的相关性，这也加速了训练的收敛。

A.4 MORE IMPLEMENTATION DETAILS

迭代边界框优化。在此，每个解码器层都会基于来自上一层的预测来优化边界框。假设有D个解码器层（例如，D=6），给定由（d-1）层解码器预测的归一化边界框$\hat{b}_q^{d-1}$，第d层解码器将对该框进行优化，优化后的边界框为：

A.7 NOTATIONS

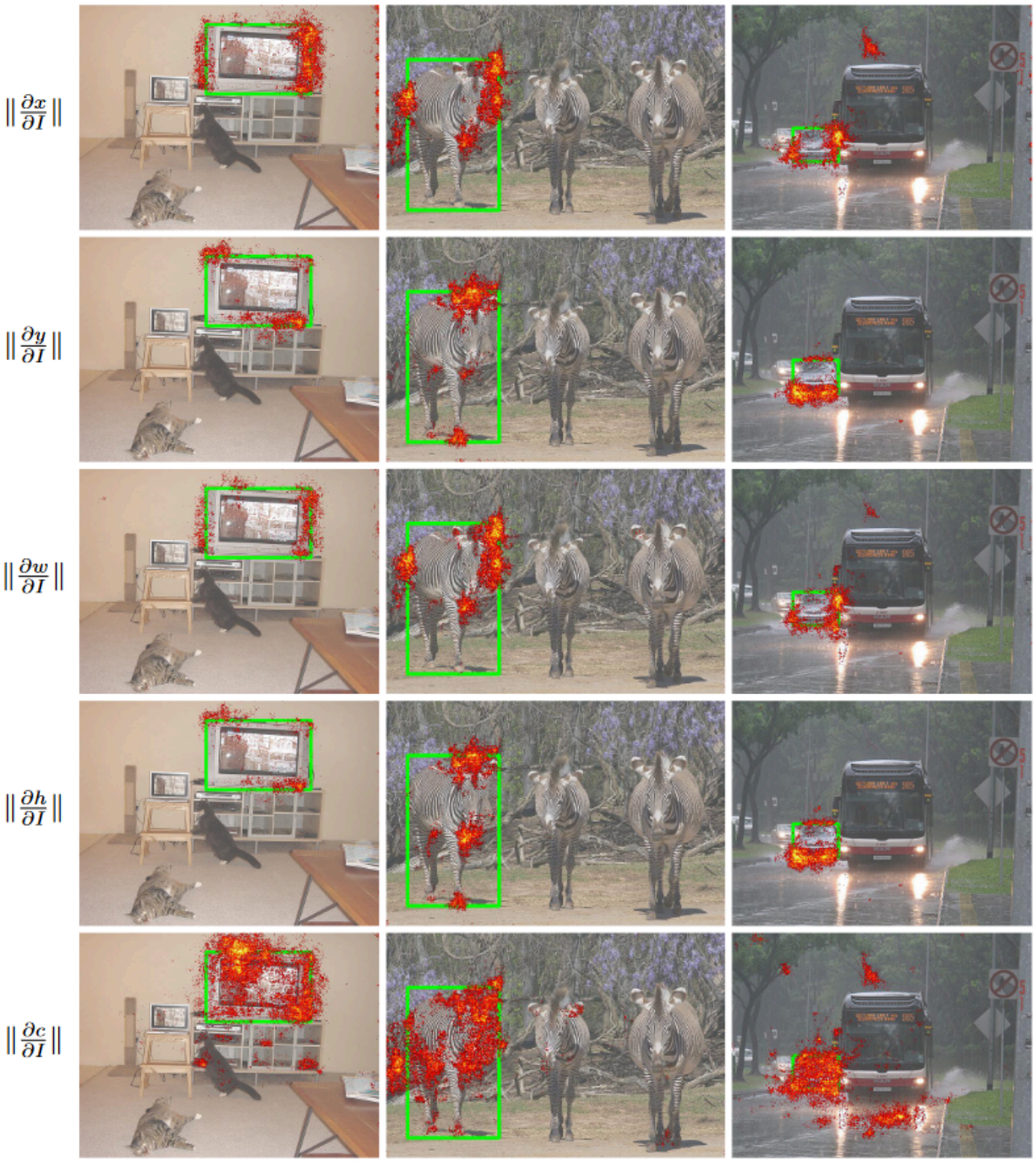

Figure 5: The gradient norm of each item (coordinate of object center (x, y), width/height of object bounding box w/h, category score c of this object) in final detection result with respect to each pixel in input image I.