

# 可变形DETR：端到端目标检测中的可变形Transformer

朱曦洲<sup>1\*</sup>, 苏伟杰<sup>2\*†</sup>, 卢乐为<sup>1</sup>, 李斌<sup>2</sup>, 王晓刚<sup>1,3</sup>, 代继峰<sup>1†</sup>  
 商汤科技研究院<sup>2</sup>中国科学技术大学<sup>3</sup>香港中文大学{zhuwalt  
 er,luotto,daijifeng}@sensetime.com jackroos@mail.ustc.edu.cn  
 , binli@ustc.edu.cn xgwang@ee.cuhk.edu.hk

## 摘要

DETR最近被提出，旨在消除目标检测中许多手工设计组件的需求，同时展现出良好的性能。然而，由于Transformer注意力模块在处理图像特征图时的局限性，它存在收敛速度慢和特征空间分辨率受限的问题。为解决这些问题，我们提出了Deformable DETR，其注意力模块仅关注参考点周围的一小组关键采样点。Deformable DETR能以比DETR少 $10\times$ 的训练周期实现更优性能（尤其是对小物体）。在COCO基准上的大量实验验证了我们方法的有效性。代码已发布于<https://github.com/fundamentalvision/Deformable-DETR>。

## 1 引言

现代目标检测器采用了大量手工设计的组件 (Liu等人, 2020)，例如锚框生成、基于规则的训练目标分配、非极大值抑制 (NMS) 后处理等，这些并非完全端到端的系统。近期，Carion等人 (2020) 提出DETR，旨在消除此类手工组件，构建了首个完全端到端的目标检测器，并取得了极具竞争力的性能。DETR采用简洁架构，结合卷积神经网络 (CNN) 与Transformer (Vaswani等人, 2017) 编码器-解码器，通过精心设计的训练信号，利用Transformer强大而通用的关系建模能力替代了手工设计的规则。

尽管DETR拥有引人注目的设计和良好的性能，但它自身也存在一些问题：(1) 与现有目标检测器相比，其收敛所需的训练周期要长得多。例如，在COCO基准测试 (Lin等人, 2014) 中，DETR需要500个周期才能收敛，这比Faster R-CNN (Ren等人, 2015) 慢了约10到20倍。(2) DETR在检测小目标时性能相对较低。现代目标检测器通常利用多尺度特征，其中小目标通过高分辨率特征图进行检测。然而，高分辨率特征图会导致DETR的计算复杂度达到难以接受的水平。上述问题主要可归因于Transformer组件在处理图像特征图时的不足。初始化时，注意力模块几乎对所有像素赋予均匀的注意力权重。需要经过长时间的训练，这些注意力权重才能学会聚焦于稀疏的有效位置。另一方面，Transformer编码器中的注意力权重计算与像素数量呈平方关系。因此，处理高分辨率特征图会带来极高的计算和内存复杂度。

在图像领域，可变形卷积 (Dai等人, 2017) 是一种强大且高效的机制，用于关注稀疏空间位置。它自然地避免了上述问题。然而，它缺乏元素关系建模机制，而这正是DETR成功的关键。

\*同等贡献。

†Corresponding author. ‡Work is done during an internship at SenseT时间研究。

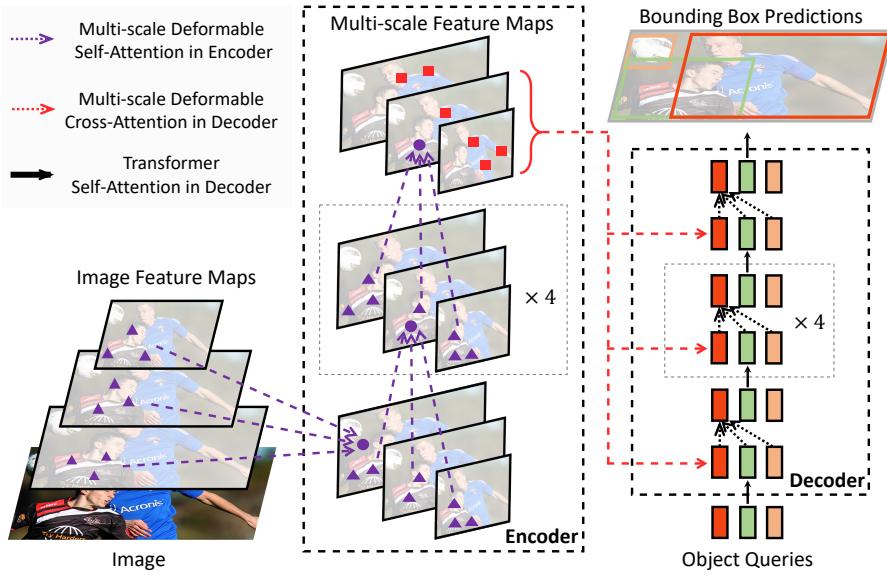


图1：所提出的可变形DETR目标检测器示意图。

本文提出了*Deformable DETR*，旨在缓解DETR收敛速度慢与复杂度高的问题。该方法融合了可变形卷积稀疏空间采样的优势与Transformer的关系建模能力。我们设计了*deformable attention module*机制，该机制通过关注少量采样位置，从所有特征图像素中筛选出关键要素作为预过滤器。该模块无需借助FPN (Lin等人, 2017a) 即可自然扩展至多尺度特征聚合。在可变形DETR中，如图1所示，我们采用（多尺度）可变形注意力模块替代原有处理特征图的Transformer注意力模块。

可变形DETR以其快速收敛性、计算高效性和内存效率，为我们探索端到端目标检测器的变体开辟了新途径。我们研究了一种简单有效的*iterative bounding box refinement*机制来提升检测性能。同时尝试了*two-stage Deformable DETR*方案，该方案中区域提议也由可变形DETR的变体生成，并进一步输入解码器进行迭代边界框优化。

在COCO基准测试 (Lin等人, 2014) 上的大量实验证明了我们方法的有效性。与DETR相比，可变形DETR能以少 $10\times$ 个训练周期达到更优性能（尤其是对小物体）。所提出的两阶段可变形DETR变体还能进一步提升性能。代码发布于<https://github.com/fundamentalvision/Deformable-DETR>。

## 2 相关工作

高效注意力机制。Transformer (Vaswani等人, 2017) 包含自注意力与交叉注意力机制。其最广为人知的局限在于当键值元素数量庞大时，高昂的时间与内存复杂度会阻碍模型扩展性 (Tay等人, 2020b)。近期研究从三大方向着手解决该问题： $\{v^*\}$ 保持原公式标记不变。

第一类方法是在键上使用预定义的稀疏注意力模式。最直接的范式是将注意力模式限制为固定的局部窗口。大多数研究 (Liu et al., 2018a; Parmar et al., 2018; Child et al., 2019; Huang et al., 2019; Ho et al., 2019; Wang et al., 2020a; Hu et al., 2019; Ramachandran et al., 2019; Qiu et al., 2019; Beltagy et al., 2020; Ainslie et al., 2020; Zaheer et al., 2020) 遵循了这一范式。尽管将注意力模式限制在局部邻域可以降低复杂度，但会丢失全局信息。为了弥补这一点，Child等人 (2019)、Huang等人 (2019)、Ho等人 (2019)、Wang等人 (2020a) 对关键元素 $\{v^*\}$ 进行了关注。

以固定间隔显著增加对键的感知范围。Beltagy等人（2020）；Ainslie等人（2020）；Zaheer等人（2020）允许少量特殊标记访问所有关键元素。Zaheer等人（2020）；Qiu等人（2019）还加入了一些预定义的稀疏注意力模式，以直接关注远处的关键元素。

第二类方法是学习数据依赖的稀疏注意力机制。Kitaev等人（2020）提出了一种基于局部敏感哈希（LSH）的注意力计算方式，通过将查询和键元素哈希到不同的桶中实现。Roy等人（2020）提出了类似思想，采用k均值聚类找出最相关的键。Tay等人（2020a）则通过学习块置换来实现分块稀疏注意力。

第三类研究旨在探索自注意力机制中的低秩特性。Wang等人（2020b）通过在尺寸维度而非通道维度上进行线性投影，减少了关键元素的数量。Katharopoulos等人（2020）与Choromanski等人（2020）则通过核化近似改写了自注意力的计算方式。

在图像领域，高效注意力机制的设计（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））仍局限于第一类方法。尽管理论复杂度有所降低，但Ramachandran等人（2019）和Hu等人（2019）承认，由于内存访问模式的固有局限，这类方法在实际实现中比具有相同浮点运算次数的传统卷积要慢得多（至少慢3×倍）。

另一方面，如Zhu等人（2019a）所述，卷积存在多种变体，例如可变形卷积（Dai等人，2017；Zhu等人，2019b）和动态卷积（Wu等人，2019），这些变体亦可视为自注意力机制。特别是，可变形卷积在图像识别任务上的运行效果和效率远优于Transformer自注意力机制。然而，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，属于第二类别。它仅关注由查询元素特征预测出的一小组固定采样点。与Ramachandran等人（2019）和Hu等人（2019）的研究不同，在相同FLOPs条件下，可变形注意力仅比传统卷积稍慢。

多尺度特征表示在目标检测中的应用。目标检测的主要难点之一在于如何有效表征尺度差异巨大的物体。现代检测器通常利用多尺度特征来应对这一挑战。作为开创性工作之一，FPN（Lin等人，2017a）提出自上而下的路径来融合多尺度特征。PANet（Liu等人，2018b）则在FPN基础上额外添加了自下而上的路径。Kong等人（2018）通过全局注意力操作整合所有尺度的特征。Zhao等人（2019）提出U型模块进行多尺度特征融合。近期，NAS-FPN（Ghiasi等人，2019）和Auto-FPN（Xu等人，2019）利用神经架构搜索自动设计跨尺度连接。Tan等人（2020）提出的BiFPN是PANet的重复简化版本。我们提出的多尺度可变形注意力模块无需依赖这些特征金字塔网络，即可通过注意力机制自然聚合多尺度特征图。

### 3 重新审视TRANSFORMER与DETR

Transformer中的多头注意力机制。Transformer（Vaswani等人，2017）是一种基于注意力机制的神经网络架构，专为机器翻译设计。给定一个查询元素（如输出句子中的目标词）和一组键元素（如输入句子中的源词）， $\{v^*\}$ 会根据衡量查询-键对兼容性的注意力权重，自适应地聚合键内容。为了让模型能够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会通过可学习的权重进行线性聚合。设 $\{v^*\} \Omega \{v^*\}$ 索引一个具有表示特征 $\{v^*\}$ 的查询元素， $\{v^*\} \Omega \{v^*\}$ 索引一个具有表示特征 $\{v^*\}$ 的键元素，其中 $\{v^*\}$ 是特征维度， $\Omega \{v^*\}$ 和 $\Omega \{v^*\}$ 分别指定查询元素和键元素的集合。那么，多头注意力特征的计算公式为

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

其中 $m$ 表示注意力头的索引， $\mathbf{W}'_m \in \mathbb{R}^{C_v \times C}$ 和 $\mathbf{W}_m \in \mathbb{R}^{C \times C_v}$ 为可学习权重（默认为 $C_v = C/M$ ）。注意力权重 $A_{mqk} \propto \exp\{\frac{\mathbf{z}_q^T \mathbf{U}_m^T \mathbf{V}_m \mathbf{x}_k}{\sqrt{C_v}}\}$ 通过 $\sum_{k \in \Omega_k} A_{mqk} = 1$ 进行归一化处理，其中 $\mathbf{U}_m, \mathbf{V}_m \in \mathbb{R}^{C_v \times C}$ 同样为可学习权重。为区分不同空间位置，表征特征 $\mathbf{z}_q$ 和 $\mathbf{x}_k$ 通常由元素内容与位置嵌入的拼接/求和构成。

Transformer存在两个已知问题。其一是模型需要较长的训练周期才能收敛。假设查询元素与键元素的数量分别为 $N_q$ 和 $N_k$ 。通常情况下，在参数初始化得当的前提下， $\mathbf{U}_m \mathbf{z}_q$ 和 $\mathbf{V}_m \mathbf{x}_k$ 服从均值为0、方差为1的分布，这会导致当 $N_k$ 较大时，注意力权重 $A_{mqk} \approx \frac{1}{N_k}$ 呈现均匀分布。该现象将造成输入特征的梯度方向模糊，因此需要延长训练周期使注意力权重能够聚焦于特定键值。在图像领域，键元素通常对应图像像素，此时 $N_k$ 可能极大，导致收敛过程极为缓慢。

另一方面，当查询和键元素数量庞大时，多头注意力的计算与内存复杂度可能极高。公式1的计算复杂度为 $O(N_q C^2 + N_k C^2 + N_q N_k C)$ 。在图像领域，查询和键元素均为像素的情况下 $N_q = N_k \gg C$ ，由于 $O(N_q N_k C)$ 的存在，复杂度主要由第三项主导。因此，多头注意力模块会随着特征图尺寸的增加而呈现二次方级的复杂度增长。

**DETR。** DETR (Carion等人, 2020年) 基于Transformer编码器-解码器架构构建，结合了一种基于集合的匈牙利损失，该损失通过二分匹配强制为每个真实边界框生成唯一预测。我们简要回顾其网络架构如下。

给定由CNN主干网络（如ResNet (He et al., 2016)）提取的输入特征图 $\mathbf{x} \in \mathbb{R}^{C \times H \times W}$ ，DETR采用标准的Transformer编码器-解码器架构，将这些输入特征图转换为一组对象查询的特征。在对象查询特征（由解码器生成）之上，添加了一个3层前馈神经网络 (FFN) 和一个线性投影层作为检测头。FFN充当回归分支，用于预测边界框坐标 $\mathbf{b} \in [0,1]^4$ ，其中 $\mathbf{b} = \{b_x, b_y, b_w, b_h\}$ 编码了归一化的框中心坐标、框高度和宽度（相对于图像尺寸）。线性投影层则作为分类分支，生成分类结果。

对于DETR中的Transformer编码器，查询和键元素均来自特征图中的像素。输入为ResNet特征图（带有编码的位置嵌入）。设 $H$ 和 $W$ 分别表示特征图的高度和宽度。自注意力机制的计算复杂度为 $O(H^2 W^2 C)$ ，其随空间尺寸呈二次方增长。

在DETR的Transformer解码器中，输入既包含来自编码器的特征图，也包括由可学习位置嵌入表示的 $N$ 个对象查询（例如 $N = 100$ 个）。解码器内设有两种注意力模块：交叉注意力模块与自注意力模块。在交叉注意力模块中，对象查询从特征图中提取特征，其中查询元素来自对象查询，而键元素则源自编码器输出的特征图。在此过程中， $N_q = N$ ， $N_k = H \times W$ ，且交叉注意力的复杂度为 $O(HWC^2 + NHWC)$ ，该复杂度随特征图空间尺寸线性增长。自注意力模块则使对象查询相互交互以捕捉其间关系，此时查询与键元素均来自对象查询集合。该模块涉及 $N_q = N_k = N$ ，其复杂度为 $O(2NC^2 + N^2C)$ 。当对象查询数量适中时，此复杂度处于可接受范围。

DETR是一种极具吸引力的目标检测设计方案，它消除了对许多手工设计组件的需求。然而，该系统也存在自身的问题。这些问题主要归因于Transformer注意力机制在处理图像特征图作为关键元素时的不足：(1) DETR在小物体检测上性能相对较低。现代目标检测器采用高分辨率特征图来更好地检测小物体。但高分辨率特征图会导致DETR的Transformer编码器中自注意力模块的计算复杂度急剧上升——该模块复杂度与输入特征图空间尺寸呈平方关系。(2) 相较于现代目标检测器，DETR需要更多训练周期才能收敛。这主要是因为处理图像特征的注意力模块难以训练。例如在初始化阶段，交叉注意力模块几乎对整个特征图进行平均关注；而训练结束时，注意力图会学习得非常稀疏，仅聚焦于目标物体 $\{v^*\}$ 。

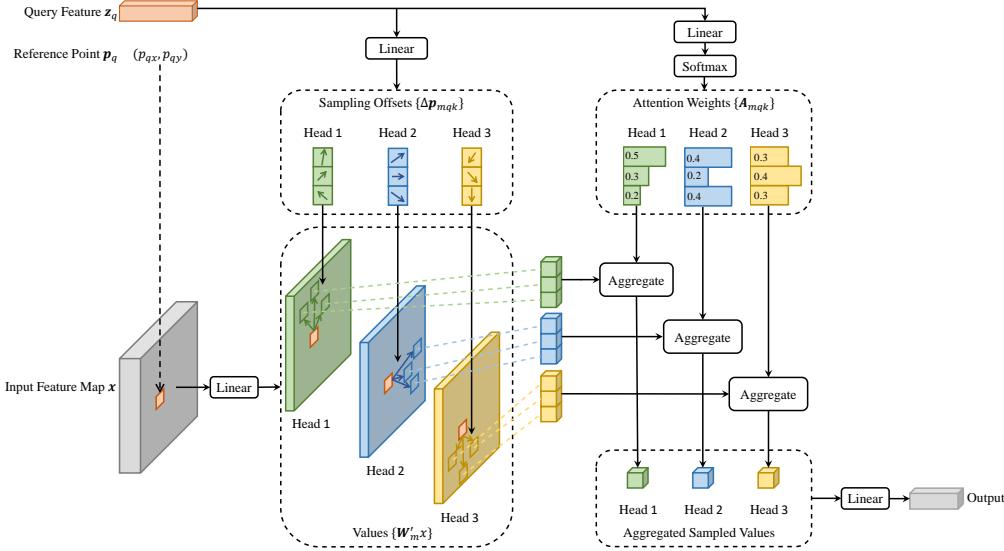


图2：所提出的可变形注意力模块示意图。

末端。似乎DETR需要一个漫长的训练周期来学习注意力图中如此显著的变化。

## 4 方法

### 4.1 用于端到端目标检测的可变形变换器

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在于，它会遍历所有可能的空间位置。为解决这一问题，我们提出了*deformable attention module*。受可变形卷积（Dai等人，2017；Zhu等人，2019b）启发，该模块仅关注参考点周围的一小组关键采样点，而忽略特征图的空间尺寸，如图2所示。通过为每个查询分配少量固定数量的键，可以缓解收敛性和特征空间分辨率的问题。

给定输入特征图  $x \in \mathbb{R}^{C \times H \times W}$ ，令  $q$  索引一个具有内容特征  $z_q$  和二维参考点  $p_q$  的查询元素，可变形注意力特征的计算方式为

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

其中  $m$  表示注意力头的索引， $k$  表示采样键的索引， $K$  为总采样键数 ( $K \ll HW$ )。

$\Delta p_{mqk}$  和  $A_{mqk}$  分别代表第  $m$  个注意力头中第  $k$  个采样点的采样偏移量和注意力权重。标量注意力权重  $A_{mqk}$  的范围在 [0, 1] 之间，通过  $\sum_{k=1}^K A_{mqk} = 1$  进行归一化处理。 $\Delta p_{mqk} \in \mathbb{R}^2$  为二维实数，其范围不受限制。由于  $p_q + \Delta p_{mqk}$  为分数形式，计算  $x(p_q + \Delta p_{mqk})$  时采用了 Dai 等人(2017)中的双线性插值法。 $\Delta p_{mqk}$  和  $A_{mqk}$  均通过对查询特征  $z_q$  进行线性投影获得。具体实现中，查询特征  $z_q$  被输入到一个  $3MK$  通道的线性投影算子，其中前  $2MK$  个通道用于编码采样偏移量  $\Delta p_{mqk}$ ，剩余  $MK$  个通道则输入 softmax 算子以生成注意力权重  $A_{mqk}$ 。

可变形注意力模块专为处理卷积特征图作为关键元素而设计。设  $N_q$  为查询元素的数量，当  $MK$  相对较小时，可变形注意力模块的复杂度为  $O(2N_q C^2 + \min(HWC^2, N_q KC^2))$ （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  $O(HWC^2)$ ，与空间尺寸呈线性关系。当作为交叉注意力模块应用时

在DETR解码器中，其中 $N_q = N$  ( $N$ 是对象查询)的数量，复杂度变为 $O(NKC^2)$ ，这与空间大小 $HW$ 无关。

多尺度可变形注意力模块。大多数现代物体检测框架得益于多尺度特征图（Liu等人，2020年）。我们提出的可变形注意力模块可以自然地扩展到多尺度特征图的应用中。

设 $\{\mathbf{x}^l\}_{l=1}^L$ 为输入的多尺度特征图，其中 $\mathbf{x}^l \in \mathbb{R}^{C \times H_l \times W_l}$ 。令 $\hat{\mathbf{p}}_q \in [0, 1]^2$ 表示每个查询元素 $q$ 参考点的归一化坐标，则应用多尺度可变形注意力模块如下

$$\text{MSDeformAttn}(\mathbf{z}_q, \hat{\mathbf{p}}_q, \{\mathbf{x}^l\}_{l=1}^L) = \sum_{m=1}^M \mathbf{W}_m \left[ \sum_{l=1}^L \sum_{k=1}^K A_{mlqk} \cdot \mathbf{W}'_m \mathbf{x}^l (\phi_l(\hat{\mathbf{p}}_q) + \Delta \mathbf{p}_{mlqk}) \right], \quad (3)$$

其中 $m$ 索引注意力头， $l$ 索引输入特征层级， $k$ 索引采样点。 $\Delta \mathbf{p}_{mlqk}$ 和 $A_{mlqk}$ 分别表示第 $l^{\text{th}}$ 特征层级中第 $m^{\text{th}}$ 注意力头的第 $k^{\text{th}}$ 个采样点的采样偏移量与注意力权重。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。此处，为清晰表达尺度关系，我们采用归一化坐标 $\hat{\mathbf{p}}_q \in [0, 1]^2$ ，其中归一化坐标(0,0)和(1,1)分别对应图像的左上角与右下角。公式3中的函数 $\phi_l(\hat{\mathbf{p}}_q)$ 将归一化坐标 $\hat{\mathbf{p}}_q$ 重新缩放到第 $l$ 层级的输入特征图上。多尺度可变形注意力机制与先前的单尺度版本极为相似，区别仅在于其从多尺度特征图中采样 $LK$ 个点，而非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 $\mathbf{W}'_m \in \mathbb{R}^{C_v \times C}$ 固定为单位矩阵时，将退化为可变形卷积（Dai等人，2017）。可变形卷积专为单尺度输入设计，每个注意力头仅聚焦于单一采样点。然而，我们的多尺度可变形注意力机制能够同时关注来自多尺度输入的多个采样点。该（多尺度）可变形注意力模块亦可视为Transformer注意力机制的高效变体，其中通过可变形采样位置引入了预过滤机制。当采样点遍历所有可能位置时，所提出的注意力模块即等同于Transformer注意力。

可变形Transformer编码器。我们将DETR中处理特征图的Transformer注意力模块替换为提出的多尺度可变形注意力模块。编码器的输入与输出均为具有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 $C_3$ 至 $C_5$ 阶段输出特征图（通过 $1 \times 1$ 卷积转换）提取多尺度特征图 $\{\mathbf{x}^l\}_{l=1}^{L-1}$  ( $L = \text{至}$ )，其中 $C_l$ 的分辨率比输入图像低 $2^{l-1}$ 倍。最低分辨率特征图 $\mathbf{x}^L$ 是通过在最后的 $C_5$ 阶段应用 $3 \times 3$ 步长2卷积获得，记为 $C_6$ 。所有多尺度特征图均具有 $C = 256$ 通道。需要注意的是，我们未采用FPN（Lin等人，2017a）中的自上而下结构，因为所提出的多尺度可变形注意力本身就能实现多尺度特征图间的信息交互。多尺度特征图的构建过程亦在附录A.2中图示说明。第5.2节的实验表明，添加FPN并不会提升性能。

在编码器中应用多尺度可变形注意力模块时，输出的是与输入分辨率相同的多尺度特征图。键元素和查询元素均来自这些多尺度特征图中的像素。对于每个查询像素，其参考点即为自身。为了识别每个查询像素所处的特征层级，除了位置嵌入外，我们还在特征表示中添加了尺度层级嵌入，记为 $\mathbf{e}_l$ 。与固定编码的位置嵌入不同，尺度层级嵌入 $\{\mathbf{e}_l\}_{l=1}^L$ 是随机初始化并与网络联合训练的。

可变形Transformer解码器。解码器中包含交叉注意力和自注意力模块。这两类注意力模块的查询元素均为目标查询。在交叉注意力模块中，目标查询从特征图中提取特征，其关键元素来自编码器输出的特征图；而在自注意力模块中，目标查询相互交互，其关键元素即为目标查询本身。由于我们提出的可变形注意力模块专为处理卷积特征图作为关键元素而设计，因此仅将每个交叉注意力模块替换为多尺度可变形注意力模块，同时保持自注意力模块不变。对于每个目标查询，其二维归一化坐标的 $\{v^*\}$

参考点 $\hat{p}_q$ 通过其对象查询嵌入预测得出，经由一个可学习的线性投影层及随后的sigmoid函数处理。

由于多尺度可变形注意力模块围绕参考点提取图像特征，我们让检测头预测边界框相对于参考点的相对偏移量，以进一步降低优化难度。参考点被用作框中心的初始猜测。检测头预测相对于参考点的相对偏移量，详情请参阅附录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，我们建立了一个高效且快速收敛的检测系统，称为可变形DETR（见图1）。

#### 4.2 可变形DETR的额外改进与变体

可变形DETR以其快速收敛性、计算高效性和内存效率，为我们探索各种端到端物体检测器的变体开辟了可能性。由于篇幅限制，这里仅介绍这些改进与变体的核心思想，具体实现细节详见附录A.4。

迭代边界框优化。这一思路受到光流估计中迭代优化技术的启发（Teed & Deng, 2020）。我们构建了一个简单高效的迭代边界框优化机制来提升检测性能。其中，每个解码器层都会基于前一层的预测结果对边界框进行精细化调整。

两阶段可变形DETR。在原始DETR中，解码器的对象查询与当前图像无关。受两阶段目标检测器的启发，我们探索了一种可变形DETR的变体，首先生成区域提议作为第一阶段。生成的区域提议将作为对象查询输入解码器进行进一步优化，从而形成一个两阶段的可变形DETR框架。

在第一阶段，为了获得高召回率的候选框，多尺度特征图中的每个像素都将作为对象查询。然而，直接将对象查询设置为像素会给解码器中的自注意力模块带来难以承受的计算和内存开销，因为其复杂度随查询数量呈二次方增长。为避免这一问题，我们移除了解码器，构建了一个仅含编码器的可变形DETR模型用于区域提议生成。在该模型中，每个像素被指定为一个对象查询，直接预测一个边界框。得分最高的边界框被选作区域提议。在将区域提议输入第二阶段前，不应用非极大值抑制(NMS)。

## 5 实验

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

实现细节。在消融实验中，我们采用ImageNet（Deng等人，2009）预训练的ResNet-50（He等人，2016）作为主干网络，并在不使用FPN（Lin等人，2017a）的情况下提取多尺度特征图。默认情况下，可变形注意力的参数设置为 $M = 8$ 和 $K = 4$ 。可变形Transformer编码器的参数在不同特征层级间共享。其余超参数设置与训练策略主要遵循DETR（Carion等人，2020），不同之处在于：边界框分类采用损失权重为2的Focal Loss（Lin等人，2017b），并将目标查询数量从100增至300。为公平比较，我们还报告了采用相同改进的DETR-DC5性能，记为DETR-DC5<sup>+</sup>。默认情况下，模型训练50个周期，学习率在第40个周期时衰减为原来的0.1倍。遵循DETR（Carion等人，2020），我们使用Adam优化器（Kingma & Ba, 2015）进行训练，基础学习率为 $2 \times 10^{-4}$ 、 $\beta_1 = 0.9$ 、 $\beta_2 = 0.999$ ，权重衰减为 $10^{-4}$ 。用于预测目标查询参考点与采样偏移量的线性投影层学习率乘以0.1系数。运行时间在NVIDIA Tesla V100 GPU上评估。

### 5.1 与DETR的对比

如表1所示，与Faster R-CNN + FPN相比，DETR需要更多的训练周期才能收敛，且在检测小物体时表现较差。相较于

DETR和Deformable DETR以少 $10\times$ 倍的训练周期实现了更优性能（尤其在小物体检测上）。具体收敛曲线如图3所示。借助迭代边界框优化及两阶段策略，我们的方法能进一步提升检测精度。

我们提出的Deformable DETR在FLOPs上与Faster R-CNN + FPN和DETR-DC5相当。但运行速度比DETR-DC5快得多（ $1.6\times$ ），仅比Faster R-CNN + FPN慢25%。DETR-DC5的速度问题主要源于Transformer注意力机制中的大量内存访问。我们提出的可变形注意力机制能够缓解这一问题，代价是无序内存访问。因此，它仍比传统卷积稍慢。

表1：Deformable DETR与DETR在COCO 2017验证集上的对比。DETR-DC5<sup>+</sup>表示采用Focal Loss和300个目标查询的DETR-DC5。

Method	Epochs	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP <sub>S</sub>	AP <sub>M</sub>	AP <sub>L</sub>	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 <sup>+</sup>	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

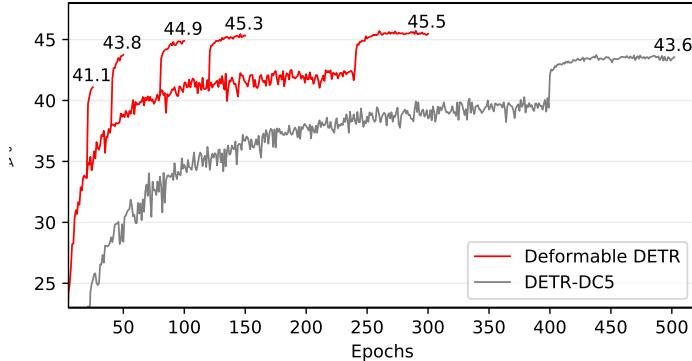


图3：Deformable DETR与DETR-DC5在COCO 2017验证集上的收敛曲线。对于Deformable DETR，我们通过改变学习率下降的周期（即AP分数跃升的节点）来探索不同的训练方案。

## 5.2 可变形注意力的消融研究

表2展示了所提出的可变形注意力模块在不同设计选择下的消融实验结果。采用多尺度输入而非单尺度输入能有效提升检测精度，带来1.7%的平均精度（AP）提升，尤其对小目标检测效果显著，提升了2.9%的AP<sub>S</sub>。增加采样点数量K可进一步带来0.9%的AP提升。采用支持跨尺度信息交互的多尺度可变形注意力机制，还能额外获得1.5%的AP提升。由于已采用跨层级特征交换机制，添加特征金字塔网络（FPN）并不会带来性能提升。当不应用多尺度注意力且K = 取值为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，导致检测精度显著下降。

## 5.3 与最先进的方法的比较

表3将所提方法与其他先进方法进行了对比。我们的模型在表3中同时采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，我们的方法在不添加额外技巧的情况下分别达到了48.7 AP和49.0 AP。通过采用结合DCN的ResNeXt-101（Zhu等人，2019b），准确率提升至50.1 AP。在引入额外测试时数据增强后，所提方法实现了52.3 AP。

表2: COCO 2017验证集上可变形注意力的消融实验。“MS输入”表示使用多尺度输入。“MS注意力”表示使用多尺度可变形注意力。 $K$ 为每个特征层级上每个注意力头的采样点数。

MS inputs	MS attention	K	FPNs	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP <sub>S</sub>	AP <sub>M</sub>	AP <sub>L</sub>
✓	✓	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		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	w/o	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

表3: Deformable DETR与最先进方法在COCO 2017测试开发集上的对比。“TTA”表示测试时数据增强，包括水平翻转和多尺度测试。

Method	Backbone	TTA	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP <sub>S</sub>	AP <sub>M</sub>	AP <sub>L</sub>
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 结论

可变形DETR是一种高效且快速收敛的端到端目标检测器。它使我们能够探索更多有趣且实用的端到端目标检测器变体。可变形DETR的核心在于（多尺度）可变形注意力模块，这是一种处理图像特征图的高效注意力机制。我们希望这项工作能为探索端到端目标检测开辟新的可能性。

致谢

该工作得到了国家重点研发计划（2020AAA0105200）、北京智源人工智能研究院以及国家自然科学基金（项目编号U19B2044和61836011）的资助。

## 参考文献

约书亚·安斯利、圣地亚哥·翁塔农、克里斯·阿尔贝蒂、菲利普·范、阿尼鲁德·拉武拉和苏米特·桑海。ETC: 在Transformer中编码长结构化数据。*arXiv preprint arXiv:2004.08483*, 2020年。

Iz Beltagy、Matthew E Peters 和 Arman Cohan。《Longformer: 长文档Transformer》。*arXiv preprint arXiv:2004.05150*, 2020年。

尼古拉斯·卡里昂、弗朗西斯科·马萨、加布里埃尔·辛纳夫、尼古拉斯·乌松尼尔、亚历山大·基里洛夫和谢尔盖·扎戈鲁伊科。基于Transformer的端到端目标检测。发表于ECCV, 2020年。

Rewon Child、Scott Gray、Alec Radford与Ilya Sutskever。使用稀疏变换器生成长序列。*arXiv preprint arXiv:1904.10509*, 2019年。

Krzysztof Choromanski、Valerii Likhoshesterov、David Dohan、Xingyou Song、Jared Davis、Tamas Sarlos、David Belanger、Lucy Colwell和Adrian Weller。通过线性可扩展长上下文变换器实现蛋白质的掩码语言建模。*arXiv preprint arXiv:2006.03555*, 2020年。

代继峰、齐浩志、熊宇文、李毅、张国栋、胡涵和韦毅晨。可变形卷积网络。发表于ICCV，2017年。

贾登、董伟、Richard Socher、李立佳、李凯和Fei-Fei Li。ImageNet：一个大规模层次化图像数据库。载于CVPR，2009年。

高纳兹·加西（Golnaz Ghiasi）、林宗毅（Tsung-Yi Lin）与黎国维（Quoc V. Le）。NAS-FPN：学习可扩展特征金字塔架构用于目标检测。发表于CVPR，2019年。

何恺明、张翔宇、任少卿、孙剑。深度残差学习在图像识别中的应用。发表于CVPR，2016年。

乔纳森·何、纳尔·卡尔奇布伦纳、德克·魏森伯恩与蒂姆·萨利曼斯。《多维变换器中的轴向注意力机制》。arXiv preprint arXiv:1912.12180, 2019年。

韩虎、张正、谢振达和林史蒂芬。局部关系网络在图像识别中的应用。发表于ICCV，2019年。

黄子龙、王兴刚、黄立超、黄畅、魏云超和刘文予。CCNet：语义分割中的交叉注意力机制。发表于ICCV，2019年。

Angelos Katharopoulos、Apoorv Vyas、Nikolaos Pappas 和 Francois Fleuret。Transformer 即 RNN：具有线性注意力的快速自回归 Transformer。arXiv preprint arXiv:2006.16236, 2020年。

Diederik P Kingma 和 Jimmy Ba。Adam：一种随机优化方法。发表于ICLR，2015年。

尼基塔·基塔耶夫、乌卡什·凯泽与安瑟姆·列夫斯卡娅。《Reformer：高效Transformer》。载于ICLR，2020年。

陶琨、孙富春、谭传奇、刘华平、黄文炳。面向目标检测的深度特征金字塔重配置。载于ECCV，2018年。

Tsung-Yi Lin、Michael Maire、Serge Belongie、James Hays、Pietro Perona、Deva Ramanan、Piotr Dollár 和 C Lawrence Zitnick。Microsoft COCO：上下文中的常见物体。收录于ECCV，2014年。

林惊毅（Tsung-Yi Lin）、Piotr Dollár、Ross Girshick、何恺明（Kaiming He）、Bharath Hariharan与Serge Belongie。特征金字塔网络在目标检测中的应用。载于CVPR，2017a。

林宗仪、Priya Goyal、Ross Girshick、何恺明和Piotr Dollár。密集目标检测中的焦点损失。载于ICCV，2017b。

李刘、欧阳万里、王晓刚、Paul Fieguth、陈杰、刘新旺，以及Matti Pietikäinen。深度学习在通用目标检测中的应用综述。IJCV，2020年。

彼得·J·刘、穆罕默德·萨利赫、艾蒂安·波特、本·古德里奇、瑞安·塞帕西、卢卡什·凯泽尔和诺姆·沙泽尔。通过总结长序列生成维基百科。载于ICLR，2018a。

刘澍、齐璐、秦海芳、石建萍、贾佳亚。面向实例分割的路径聚合网络。发表于CVPR，2018b。

Niki Parmar、Ashish Vaswani、Jakob Uszkoreit、ukasz Kaiser、Noam Shazeer、Alexander Ku 和 Dustin Tran。图像变换器。发表于ICML，2018年。

邱杰中、马浩、Omer Levy、Yih Wen-tau Scott、王思农和唐杰。分块自注意力机制在长文档理解中的应用。arXiv preprint arXiv:1911.02972, 2019年。

Prajit Ramachandran、Niki Parmar、Ashish Vaswani、Irwan Bello、Anselm Levskaya与Jonathon Shlens。视觉模型中的独立自注意力机制。发表于NeurIPS，2019年。

邵庆仁、何恺明、Ross Girshick和孙剑。Faster R-CNN：利用区域提议网络实现实时目标检测。发表于NeurIPS，2015年。

Aurko Roy、Mohammad Saffar、Ashish Vaswani 和 David Grangier。基于内容的高效稀疏注意力机制：路由变换器。arXiv preprint arXiv:2003.05997, 2020年。

宋光鲁、刘宇和王晓刚。重探目标检测器中的兄弟头结构。载于*CVPR*, 2020年。

Mingxing Tan、Ruoming Pang与Quoc V Le。EfficientDet：可扩展且高效的目标检测。发表于*CVPR*, 2020年。

Yi Tay、Dara Bahri、刘阳、Donald Metzler 和 Da-Cheng Juan。《稀疏Sinkhorn注意力》。载于*ICML*, 2020a。

Yi Tay、Mostafa Dehghani、Dara Bahri 和 Donald Metzler。高效Transformer：综述。*arXiv preprint arXiv:2009.06732*, 2020b。

扎卡里·蒂德与邓嘉。RAFT：用于光流的循环全对场变换。载于*ECCV*, 2020年。

田志、沈春华、陈浩和何通。FCOS：全卷积一阶段目标检测。发表于*ICCV*, 2019年。

阿希什·瓦斯瓦尼 (Ashish Vaswani)、诺姆·沙泽尔 (Noam Shazeer)、尼基·帕尔马 (Niki Parmar)、雅各布·乌兹科雷特 (Jakob Uszkoreit)、利昂·琼斯 (Llion Jones)、艾丹·N·戈麦斯 (Aidan N Gomez)、卢卡什·凯泽尔 (Łukasz Kaiser) 和伊利亚·波洛苏金 (Illia Polosukhin) 智慧注意力机制就是你所需要的。Huang, Yan, NeurIPS 2018 陈良杰。Axial-DeepLab：用于全景分割的独立轴向注意力机制。*arXiv preprint arXiv:2003.07853*, 2020a。

王思农，李贝琳达，马迪安·哈布萨，方涵，与马浩。Lformer：线性复杂度的自注意力机制。*arXiv preprint arXiv:2006.04768*, 2020b。

费利克斯·吴、安吉拉·范、阿列克谢·巴耶夫斯基、杨立昆与迈克尔·奥利。采用轻量级动态卷积减少注意力机制开销。发表于*ICLR*, 2019年。

谢赛宁、罗斯·吉斯克、皮奥特·多拉尔、屠卓文和何恺明。深度神经网络的聚合残差变换。载于*CVPR*, 2017年。

徐航、姚乐炜、张伟、梁晓丹和李振国。Auto-FPN：超越分类的物体检测网络架构自动适配。于*ICCV*, 2019年。

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, 等. Big bird: 面向更长序列的Transformer模型。*arXiv preprint arXiv:2007.14062*, 2020.

张世峰、迟程、姚永强、雷震和李Stan Z。通过自适应训练样本选择弥合基于锚点与无锚点检测之间的差距。发表于*CVPR*, 2020年。

赵启杰、盛涛、王永涛、唐智、陈颖、蔡玲与凌海滨。M2Det：基于多级特征金字塔网络的单次目标检测器。发表于*AAAI*, 2019年。

朱锡洲、程大治、张正、林史蒂芬和戴继峰。深度网络中空间注意力机制的实证研究。见 *ICCV*, 2019a。

朱锡洲、胡翰、林史蒂芬和戴继峰。可变形卷积网络v2：更灵活，效果更佳。发表于*CVPR*, 2019b。

## 附录

### A.1 可变形注意力的复杂度

假设查询元素的数量为  $N_q$ ，在可变形注意力模块中（见公式2），计算采样坐标偏移  $\Delta p_{mqk}$  和注意力权重  $A_{mqk}$  的复杂度为  $O(3N_q CMK)$ 。给定采样坐标偏移和注意力权重后，计算公式2的复杂度为  $O(N_q C^2 + N_q KC^2 + 5N_q KC)$ ，其中  $5N_q KC$  的因子源于双线性插值和注意力中的加权求和。另一方面，我们也可以在采样前计算  $\mathbf{W}'_m \mathbf{x}$ ，因为它与查询无关，此时计算公式2的复杂度将变为  $O(N_q C^2 + HWC^2 + 5N_q KC)$ 。因此，可变形注意力的总体复杂度为  $O(N_q C^2 + \min(HWC^2, N_q KC^2) + 5N_q KC + 3N_q CMK)$ 。在我们的实验中，默认设置  $M = 8$ 、 $K \leq 4$  和  $C = 256$ ，故  $5K + 3MK < C$ ，复杂度为  $O(2N_q C^2 + \min(HWC^2, N_q KC^2))$ 。

### A.2 为可变形DETR构建多尺度特征图

如第4.1节所述并如图4所示，编码器的输入多尺度特征图  $\{\mathbf{x}^l\}_{l=1}^{L-1}$  ( $L = 5$ ) 提取自 ResNet (He 等人, 2016) 中  $C_3$  至  $C_5$  阶段的输出特征图（通过  $1 \times 1$  卷积变换得到）。最低分辨率特征图  $\mathbf{x}^L$  是通过在最终  $C_5$  阶段应用  $3 \times 3$  步长 2 的卷积获得的。需要注意的是，我们未采用 FPN (Lin 等人, 2017a)，因为所提出的多尺度可变形注意力机制本身就能实现多尺度特征图间的信息交互。

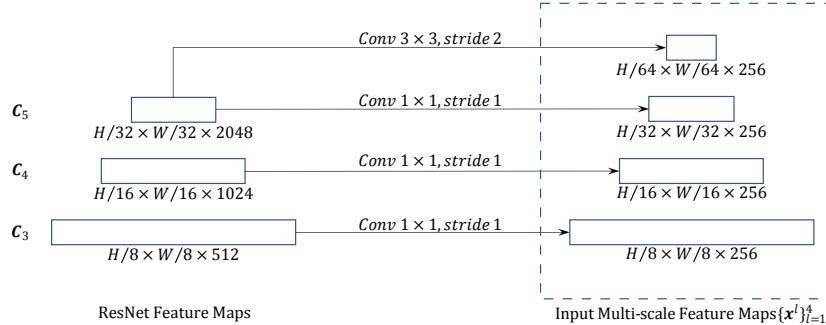


图4：为可变形DETR构建多尺度特征图。

### A.3 可变形DETR中的边界框预测

由于多尺度可变形注意力模块围绕参考点提取图像特征，我们设计检测头以预测相对于参考点的边界框偏移量，从而进一步降低优化难度。参考点被用作框中心的初始猜测。检测头预测相对于参考点的相对偏移量  $\hat{\mathbf{p}}_q = (\hat{p}_{qx}, \hat{p}_{qy})$ ，即  $\hat{\mathbf{b}}_q = \{\sigma(b_{qx} + \sigma^{-1}(\hat{p}_{qx})), \sigma(b_{qy} + \sigma^{-1}(\hat{p}_{qy})), \sigma(b_{qw}), \sigma(b_{qh})\}$ ，其中  $b_{q\{x,y,w,h\}} \in \mathbb{R}$  由检测头预测得出。 $\sigma$  和  $\sigma^{-1}$  分别表示 sigmoid 函数和反 sigmoid 函数。使用  $\sigma$  和  $\sigma^{-1}$  的目的是确保  $\hat{\mathbf{b}}$  为归一化坐标，因为  $\hat{\mathbf{b}}_q \in [0, 1]^4$ 。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相关性，这也加速了训练收敛。

### A.4 更多实现细节

迭代式边界框优化。这里，每个解码器层基于前一层的预测结果对边界框进行优化。假设共有  $D$  个解码器层（例如  $D = 6$ ），给定由第  $d - 1$  层解码器预测的归一化边界框  $\hat{\mathbf{b}}_q^{d-1}$ ，第  $d$  层

解码器层将框细化为

$$\hat{\mathbf{b}}_q^d = \{\sigma(\Delta b_{qx}^d + \sigma^{-1}(\hat{b}_{qx}^{d-1})), \sigma(\Delta b_{qy}^d + \sigma^{-1}(\hat{b}_{qy}^{d-1})), \sigma(\Delta b_{qw}^d + \sigma^{-1}(\hat{b}_{qw}^{d-1})), \sigma(\Delta b_{qh}^d + \sigma^{-1}(\hat{b}_{qh}^{d-1}))\},$$

其中  $d \in \{1, 2, \dots, D\}$ 、 $\Delta b_{q\{x,y,w,h\}}^d \in \mathbb{R}$  在第  $d$  个解码器层被预测。不同解码器层的预测头不共享参数。初始框设置为  $\hat{b}_{qx}^0 = \hat{p}_{qx}$ 、 $\hat{b}_{qy}^0 = \hat{p}_{qy}$ 、 $\hat{b}_{qw}^0 = 0.1$ ，以及  $\hat{b}_{qh}^0 = 0.1$ 。该系统对  $b_{qw}^0$  和  $b_{qh}^0$  的选择具有鲁棒性。我们尝试将其设为 0.05、0.1、0.2、0.5，均获得了相似的性能。为稳定训练，与 Teed & Deng (2020) 类似，梯度仅通过  $\Delta b_{q\{x,y,w,h\}}^d$  反向传播，并在

$$\sigma^{-1}(\hat{b}_{q\{x,y,w,h\}}^{d-1}).$$

在迭代边界框优化过程中，对于第  $d$  个解码器层，我们采样与第  $d-1$  个解码器层预测的边界框  $\hat{b}_q^{d-1}$  相对应的关键元素。在第  $d$  个解码器层交叉注意力模块的公式 3 中， $\hat{b}_{qx}^{d-1}, \hat{b}_{qy}^{d-1}$  作为新的参考点。采样偏移量  $\Delta p_{mlqk}$  同样受到边界框尺寸的调制，如  $\Delta p_{mlqkx} \hat{b}_{qw}^{d-1}, \Delta p_{mlqky} \hat{b}_{qh}^{d-1}$  所示。这些调整使得采样位置与先前预测边界框的中心和尺寸相关联。

两阶段可变形 DETR。在第一阶段，给定编码器输出的特征图，对每个像素点应用检测头。该检测头分别采用三层 FFN 进行边界框回归，以及线性投影完成边界框二分类（即前景与背景）。设  $i$  表示来自特征层级  $l_i \in \{1, 2, \dots, L\}$  的像素索引，其二维归一化坐标为  $\hat{\mathbf{p}}_i = (\hat{p}_{ix}, \hat{p}_{iy}) \in [0, 1]^2$ ，其对应边界框由以下公式预测得出：

$$\hat{\mathbf{b}}_i = \{\sigma(\Delta b_{ix} + \sigma^{-1}(\hat{p}_{ix})), \sigma(\Delta b_{iy} + \sigma^{-1}(\hat{p}_{iy})), \sigma(\Delta b_{iw} + \sigma^{-1}(2^{l_i-1}s)), \sigma(\Delta b_{ih} + \sigma^{-1}(2^{l_i-1}s))\},$$

其中基础物体尺度  $s$  设为 0.05， $\Delta b_{i\{x,y,w,h\}} \in \mathbb{R}$  由边界框回归分支预测。DETR 中的匈牙利损失用于训练检测头。

给定第一阶段预测的边界框，得分最高的边界框被选为区域提议。在第二阶段，这些区域提议作为初始框输入解码器，用于 *iterative bounding box refinement*，其中对象查询的位置嵌入被设置为区域提议坐标的位置嵌入。

多尺度可变形注意力的初始化。在我们的实验中，注意力头的数量设为  $M = 8$ 。在多尺度可变形注意力模块中， $\mathbf{W}'_m \in \mathbb{R}^{C_v \times C}$  和  $\mathbf{W}_m \in \mathbb{R}^{C \times C_v}$  被随机初始化。用于预测  $A_{mlqk}$  和  $\Delta p_{mlqk}$  的线性投影权重参数初始化为零。线性投影的偏置参数初始化时，使  $A_{mlqk} = \frac{1}{LK}$  和  $\{\Delta p_{1lqk} = (-k, -k), \Delta p_{2lqk} = (-k, 0), \Delta p_{3lqk} = (-k, k), \Delta p_{4lqk} = (0, -k), \Delta p_{5lqk} = (0, k), \Delta p_{6lqk} = (k, -k), \Delta p_{7lqk} = (k, 0), \Delta p_{8lqk} = (k, k)\} (k \in \{1, 2, \dots, K\})$ 。

对于 *iterative bounding box refinement*，解码器中用于  $\Delta p_{mlqk}$  预测的初始化偏置参数会进一步与  $\frac{1}{2K}$  相乘，从而确保初始化时的所有采样点都位于前一解码器层预测的对应边界框内。

### A.5 可变形 DETR 关注什么？

为了研究 Deformable DETR 在给出最终检测结果时关注哪些区域，我们绘制了最终预测中各项（即物体中心的 x/y 坐标、物体边界框的宽度/高度、该物体的类别得分）相对于图像中每个像素的梯度范数，如图 5 所示。根据泰勒定理，梯度范数能反映输出相对于像素扰动的变化程度，因此它可以展示模型在预测各项时主要依赖哪些像素。

可视化结果表明，Deformable DETR 通过关注物体的极值点来确定其边界框，这与 DETR (Carion 等人，2020 年) 中的观察结果相似。更具体地说，Deformable DETR 会关注物体左右边界以确定 x 坐标和宽度，关注上下边界以确定 y 坐标和高度。与此同时，与 DETR (Carion 等人，2020 年) 不同的是，我们的 Deformable DETR 还会关注物体内部的像素来预测其类别。

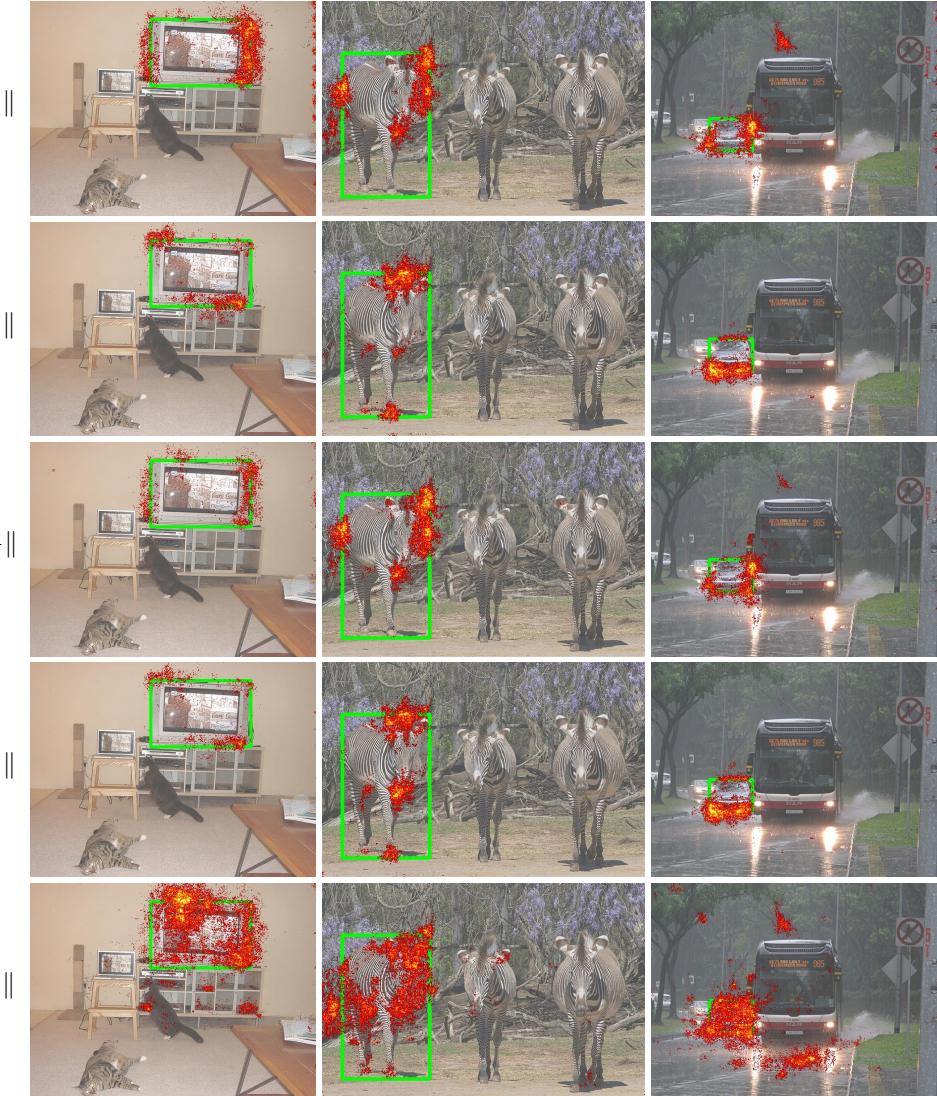


图5：最终检测结果中各项（物体中心坐标 $x, y$ 、物体边界框宽度/高度 $w/h$ 、该物体类别得分 $c$ ）相对于输入图像 $I$ 各像素的梯度范数。

#### A.6 多尺度可变形注意力的可视化

为了更好地理解学习到的多尺度可变形注意力模块，我们在图6中可视化了编码器和解码器最后一层的采样点及注意力权重。为了便于阅读，我们将来自不同分辨率特征图的采样点和注意力权重整合到一张图中。

与DETR (Carion等人, 2020年) 类似，实例在Deformable DETR的编码器中已被分离。而在解码器阶段，我们的模型关注的是整个前景实例，而非仅如DETR (Carion等人, 2020年) 所观察到的极端点。结合图5中 $\|\frac{\partial c}{\partial I}\|$ 的可视化效果，我们可以推测原因在于，我们的Deformable DETR不仅需要极端点，还需要内部点来确定物体类别。可视化结果还表明，所提出的多尺度可变形注意力模块能够根据前景对象的不同尺度和形状，自适应地调整其采样点及注意力权重。

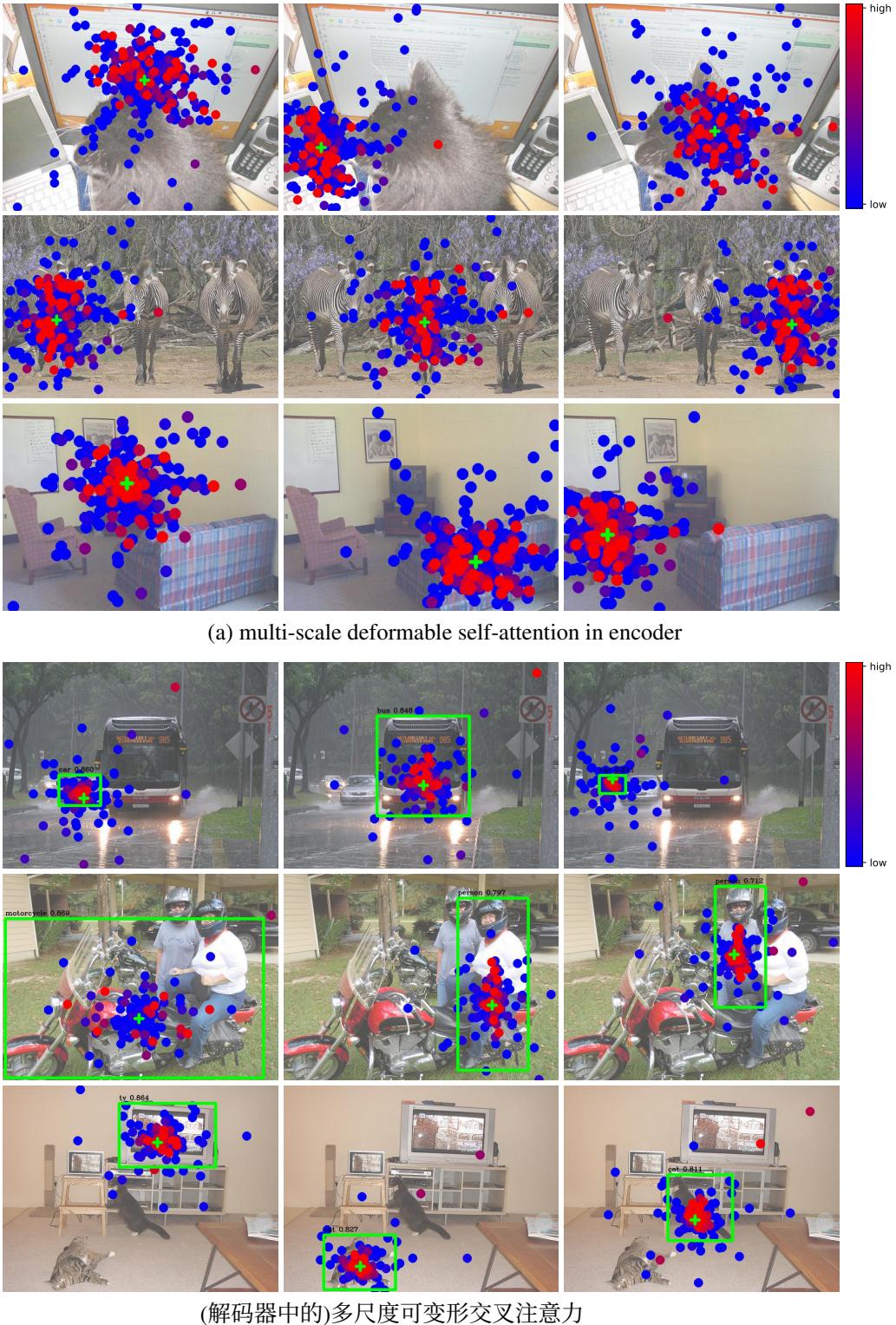


图6：多尺度可变形注意力的可视化展示。为了便于阅读，我们将不同分辨率特征图上的采样点和注意力权重绘制在同一张图中。每个采样点以实心圆点标记，其颜色代表对应的注意力权重。参考点以绿色十字标记表示，在编码器中该点也等同于查询点。解码器中，预测的边界框显示为绿色矩形框，类别及置信度分数以文字形式标注于其正上方。

## A.7 符号表示

表4：论文中符号的查找表。

Notation	Description
$m$	index for attention head
$l$	index for feature level of key element
$q$	index for query element
$k$	index for key element
$N_q$	number of query elements
$N_k$	number of key elements
$M$	number of attention heads
$L$	number of input feature levels
$K$	number of sampled keys in each feature level for each attention head
$C$	input feature dimension
$C_v$	feature dimension at each attention head
$H$	height of input feature map
$W$	width of input feature map
$H^l$	height of input feature map of $l^{th}$ feature level
$W^l$	width of input feature map of $l^{th}$ feature level
$A_{mqk}$	attention weight of $q^{th}$ query to $k^{th}$ key at $m^{th}$ head
$A_{mlqk}$	attention weight of $q^{th}$ query to $k^{th}$ key in $l^{th}$ feature level at $m^{th}$ head
$z_q$	input feature of $q^{th}$ query
$p_q$	2-d coordinate of reference point for $q^{th}$ query
$\hat{p}_q$	normalized 2-d coordinate of reference point for $q^{th}$ query
$x$	input feature map (input feature of key elements)
$x_k$	input feature of $k^{th}$ key
$x^l$	input feature map of $l^{th}$ feature level
$\Delta p_{mqk}$	sampling offset of $q^{th}$ query to $k^{th}$ key at $m^{th}$ head
$\Delta p_{mlqk}$	sampling offset of $q^{th}$ query to $k^{th}$ key in $l^{th}$ feature level at $m^{th}$ head
$W_m$	output projection matrix at $m^{th}$ head
$U_m$	input query projection matrix at $m^{th}$ head
$V_m$	input key projection matrix at $m^{th}$ head
$W'_m$	input value projection matrix at $m^{th}$ head
$\phi_l(\hat{p})$	unnormalized 2-d coordinate of $\hat{p}$ in $l^{th}$ feature level
$\exp$	exponential function
$\sigma$	sigmoid function
$\sigma^{-1}$	inverse sigmoid function