DEFORMABLE DETR: DEFORMABLE TRANSFORMERS FOR END-TO-END OBJECT DETECTION

Xizhou Zhu[1*], **Weijie Su**[2*‡], **Lewei Lu**[1], **Bin Li**[2], **Xiaogang Wang**[1,3], **Jifeng Dai**[1†]
[1]SenseTime Research
[2]University of Science and Technology of China
[3]The Chinese University of Hong Kong
{zhuwalter,luotto,daijifeng}@sensetime.com
jackroos@mail.ustc.edu.cn, binli@ustc.edu.cn
xgwang@ee.cuhk.edu.hk

ABSTRACT

DETR has been recently proposed to eliminate the need for many hand-designed components in object detection while demonstrating good performance. However, it suffers from slow convergence and limited feature spatial resolution, due to the limitation of Transformer attention modules in processing image feature maps. To mitigate these issues, we proposed Deformable DETR, whose attention modules only attend to a small set of key sampling points around a reference. Deformable DETR can achieve better performance than DETR (especially on small objects) with 10× less training epochs. Extensive experiments on the COCO benchmark demonstrate the effectiveness of our approach. Code is released at https://github.com/fundamentalvision/Deformable-DETR.

1 INTRODUCTION

Modern object detectors employ many hand-crafted components (Liu et al., 2020), e.g., anchor generation, rule-based training target assignment, non-maximum suppression (NMS) post-processing. They are not fully end-to-end. Recently, Carion et al. (2020) proposed DETR to eliminate the need for such hand-crafted components, and built the first fully end-to-end object detector, achieving very competitive performance. DETR utilizes a simple architecture, by combining convolutional neural networks (CNNs) and Transformer (Vaswani et al., 2017) encoder-decoders. They exploit the versatile and powerful relation modeling capability of Transformers to replace the hand-crafted rules, under properly designed training signals.

Despite its interesting design and good performance, DETR has its own issues: (1) It requires much longer training epochs to converge than the existing object detectors. For example, on the COCO (Lin et al., 2014) benchmark, DETR needs 500 epochs to converge, which is around 10 to 20 times slower than Faster R-CNN (Ren et al., 2015). (2) DETR delivers relatively low performance at detecting small objects. Modern object detectors usually exploit multi-scale features, where small objects are detected from high-resolution feature maps. Meanwhile, high-resolution feature maps lead to unacceptable complexities for DETR. The above-mentioned issues can be mainly attributed to the deficit of Transformer components in processing image feature maps. At initialization, the attention modules cast nearly uniform attention weights to all the pixels in the feature maps. Long training epoches is necessary for the attention weights to be learned to focus on sparse meaningful locations. On the other hand, the attention weights computation in Transformer encoder is of quadratic computation w.r.t. pixel numbers. Thus, it is of very high computational and memory complexities to process high-resolution feature maps.

In the image domain, deformable convolution (Dai et al., 2017) is of a powerful and efficient mechanism to attend to sparse spatial locations. It naturally avoids the above-mentioned issues. While it lacks the element relation modeling mechanism, which is the key for the success of DETR.

[*]Equal contribution. [†]Corresponding author. [‡]Work is done during an internship at SenseTime Research.

可变形DETR：端到端目标检测中的可变形Transformer

朱曦洲[1*]，苏伟杰[2*‡]，卢乐为[1]，李斌[2]，王晓刚[1,3]，代继峰[1†]
商汤科技研究院[2]中国科学技术大学[3]香港中文大学{zhuwalter,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×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。

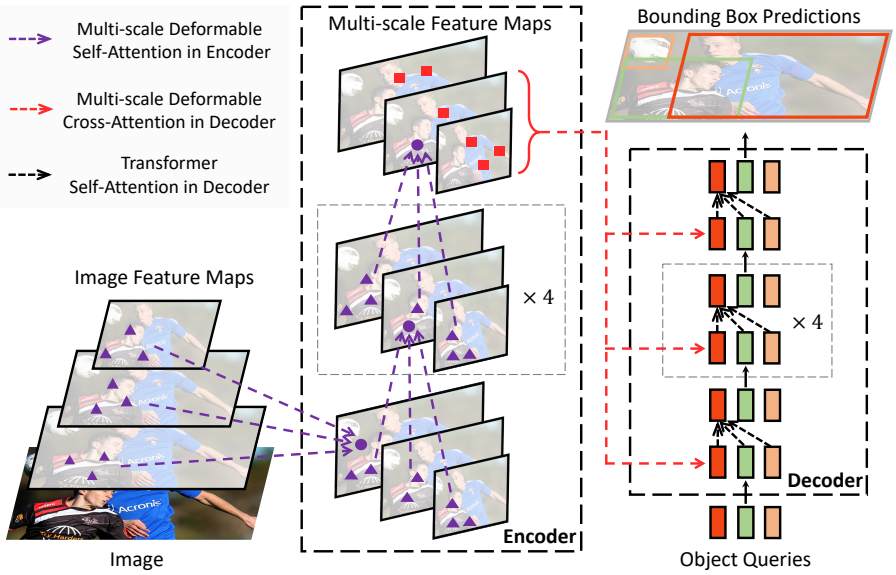


Figure 1: Illustration of the proposed Deformable DETR object detector.

In this paper, we propose *Deformable DETR*, which mitigates the slow convergence and high complexity issues of DETR. It combines the best of the sparse spatial sampling of deformable convolution, and the relation modeling capability of Transformers. We propose the *deformable attention module*, which attends to a small set of sampling locations as a pre-filter for prominent key elements out of all the feature map pixels. The module can be naturally extended to aggregating multi-scale features, without the help of FPN (Lin et al., 2017a). In Deformable DETR , we utilize (multi-scale) deformable attention modules to replace the Transformer attention modules processing feature maps, as shown in Fig. 1.

Deformable DETR opens up possibilities for us to exploit variants of end-to-end object detectors, thanks to its fast convergence, and computational and memory efficiency. We explore a simple and effective *iterative bounding box refinement* mechanism to improve the detection performance. We also try a *two-stage Deformable DETR*, where the region proposals are also generated by a vaiant of Deformable DETR, which are further fed into the decoder for iterative bounding box refinement.

Extensive experiments on the COCO (Lin et al., 2014) benchmark demonstrate the effectiveness of our approach. Compared with DETR, Deformable DETR can achieve better performance (especially on small objects) with 10× less training epochs. The proposed variant of two-stage Deformable DETR can further improve the performance. Code is released at `https://github.com/fundamentalvision/Deformable-DETR`.

2 Related Work

Efficient Attention Mechanism. Transformers (Vaswani et al., 2017) involve both self-attention and cross-attention mechanisms. One of the most well-known concern of Transformers is the high time and memory complexity at vast key element numbers, which hinders model scalability in many cases. Recently, many efforts have been made to address this problem (Tay et al., 2020b), which can be roughly divided into three categories in practice.

The first category is to use pre-defined sparse attention patterns on keys. The most straightforward paradigm is restricting the attention pattern to be fixed local windows. Most works (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) follow this paradigm. Although restricting the attention pattern to a local neighborhood can decrease the complexity, it loses global information. To compensate, Child et al. (2019); Huang et al. (2019); Ho et al. (2019); Wang et al. (2020a) attend key elements

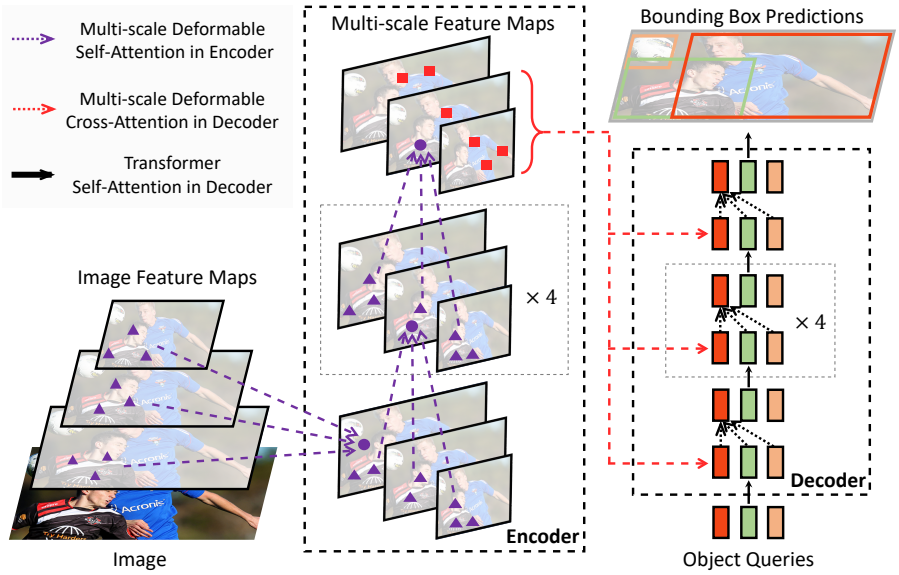


图1：所提出的可变形DETR目标检测器示意图。

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 attentionmodule*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×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。

at fixed intervals to significantly increase the receptive field on keys. Beltagy et al. (2020); Ainslie et al. (2020); Zaheer et al. (2020) allow a small number of special tokens having access to all key elements. Zaheer et al. (2020); Qiu et al. (2019) also add some pre-fixed sparse attention patterns to attend distant key elements directly.

The second category is to learn data-dependent sparse attention. Kitaev et al. (2020) proposes a locality sensitive hashing (LSH) based attention, which hashes both the query and key elements to different bins. A similar idea is proposed by Roy et al. (2020), where k-means finds out the most related keys. Tay et al. (2020a) learns block permutation for block-wise sparse attention.

The third category is to explore the low-rank property in self-attention. Wang et al. (2020b) reduces the number of key elements through a linear projection on the size dimension instead of the channel dimension. Katharopoulos et al. (2020); Choromanski et al. (2020) rewrite the calculation of self-attention through kernelization approximation.

In the image domain, the designs of efficient attention mechanism (e.g., 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)) are still limited to the first category. Despite the theoretically reduced complexity, Ramachandran et al. (2019); Hu et al. (2019) admit such approaches are much slower in implementation than traditional convolution with the same FLOPs (at least $3\times$ slower), due to the intrinsic limitation in memory access patterns.

On the other hand, as discussed in Zhu et al. (2019a), there are variants of convolution, such as deformable convolution (Dai et al., 2017; Zhu et al., 2019b) and dynamic convolution (Wu et al., 2019), that also can be viewed as self-attention mechanisms. Especially, deformable convolution operates much more effectively and efficiently on image recognition than Transformer self-attention. Meanwhile, it lacks the element relation modeling mechanism.

Our proposed deformable attention module is inspired by deformable convolution, and belongs to the second category. It only focuses on a small fixed set of sampling points predicted from the feature of query elements. Different from Ramachandran et al. (2019); Hu et al. (2019), deformable attention is just slightly slower than the traditional convolution under the same FLOPs.

Multi-scale Feature Representation for Object Detection. One of the main difficulties in object detection is to effectively represent objects at vastly different scales. Modern object detectors usually exploit multi-scale features to accommodate this. As one of the pioneering works, FPN (Lin et al., 2017a) proposes a top-down path to combine multi-scale features. PANet (Liu et al., 2018b) further adds an bottom-up path on the top of FPN. Kong et al. (2018) combines features from all scales by a global attention operation. Zhao et al. (2019) proposes a U-shape module to fuse multi-scale features. Recently, NAS-FPN (Ghiasi et al., 2019) and Auto-FPN (Xu et al., 2019) are proposed to automatically design cross-scale connections via neural architecture search. Tan et al. (2020) proposes the BiFPN, which is a repeated simplified version of PANet. Our proposed multi-scale deformable attention module can naturally aggregate multi-scale feature maps via attention mechanism, without the help of these feature pyramid networks.

3 Revisiting Transformers and DETR

Multi-Head Attention in Transformers. Transformers (Vaswani et al., 2017) are of a network architecture based on attention mechanisms for machine translation. Given a query element (e.g., a target word in the output sentence) and a set of key elements (e.g., source words in the input sentence), the *multi-head attention module* adaptively aggregates the key contents according to the attention weights that measure the compatibility of query-key pairs. To allow the model focusing on contents from different representation subspaces and different positions, the outputs of different attention heads are linearly aggregated with learnable weights. Let $q \in \Omega_q$ indexes a query element with representation feature $\boldsymbol{z}_q \in \mathbb{R}^C$, and $k \in \Omega_k$ indexes a key element with representation feature $\boldsymbol{x}_k \in \mathbb{R}^C$, where C is the feature dimension, Ω_q and Ω_k specify the set of query and key elements, respectively. Then the multi-head attention feature is calculated by

$$\text{MultiHeadAttn}(\boldsymbol{z}_q, \boldsymbol{x}) = \sum_{m=1}^{M} \boldsymbol{W}_m \big[\sum_{k \in \Omega_k} A_{mqk} \cdot \boldsymbol{W}'_m \boldsymbol{x}_k \big], \tag{1}$$

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*} Ω{v*}索引一个具有表示特征{v*}的查询元素，{v*} Ω{v*}索引一个具有表示特征{v*}的键元素，其中{v*}是特征维度，Ω{v*}和Ω{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

$$\text{MultiHeadAttn}(\boldsymbol{z}_q, \boldsymbol{x}) = \sum_{m=1}^{M} \boldsymbol{W}_m \big[\sum_{k \in \Omega_k} A_{mqk} \cdot \boldsymbol{W}'_m \boldsymbol{x}_k \big], \tag{1}$$

where m indexes the attention head, $\boldsymbol{W}'_m \in \mathbb{R}^{C_v \times C}$ and $\boldsymbol{W}_m \in \mathbb{R}^{C \times C_v}$ are of learnable weights ($C_v = C/M$ by default). The attention weights $A_{mqk} \propto \exp\{\frac{\boldsymbol{z}_q^T \boldsymbol{U}_m^T \boldsymbol{V}_m \boldsymbol{x}_k}{\sqrt{C_v}}\}$ are normalized as $\sum_{k \in \Omega_k} A_{mqk} = 1$, in which $\boldsymbol{U}_m, \boldsymbol{V}_m \in \mathbb{R}^{C_v \times C}$ are also learnable weights. To disambiguate different spatial positions, the representation features $\boldsymbol{z}_q$ and $\boldsymbol{x}_k$ are usually of the concatenation/summation of element contents and positional embeddings.

There are two known issues with Transformers. One is Transformers need long training schedules before convergence. Suppose the number of query and key elements are of N_q and N_k, respectively. Typically, with proper parameter initialization, $\boldsymbol{U}_m \boldsymbol{z}_q$ and $\boldsymbol{V}_m \boldsymbol{x}_k$ follow distribution with mean of 0 and variance of 1, which makes attention weights $A_{mqk} \approx \frac{1}{N_k}$, when N_k is large. It will lead to ambiguous gradients for input features. Thus, long training schedules are required so that the attention weights can focus on specific keys. In the image domain, where the key elements are usually of image pixels, N_k can be very large and the convergence is tedious.

On the other hand, the computational and memory complexity for multi-head attention can be very high with numerous query and key elements. The computational complexity of Eq. 1 is of $O(N_q C^2 + N_k C^2 + N_q N_k C)$. In the image domain, where the query and key elements are both of pixels, $N_q = N_k \gg C$, the complexity is dominated by the third term, as $O(N_q N_k C)$. Thus, the multi-head attention module suffers from a quadratic complexity growth with the feature map size.

DETR. DETR (Carion et al., 2020) is built upon the Transformer encoder-decoder architecture, combined with a set-based Hungarian loss that forces unique predictions for each ground-truth bounding box via bipartite matching. We briefly review the network architecture as follows.

Given the input feature maps $\boldsymbol{x} \in \mathbb{R}^{C \times H \times W}$ extracted by a CNN backbone (e.g., ResNet (He et al., 2016)), DETR exploits a standard Transformer encoder-decoder architecture to transform the input feature maps to be features of a set of object queries. A 3-layer feed-forward neural network (FFN) and a linear projection are added on top of the object query features (produced by the decoder) as the detection head. The FFN acts as the regression branch to predict the bounding box coordinates $\boldsymbol{b} \in [0, 1]^4$, where $\boldsymbol{b} = \{b_x, b_y, b_w, b_h\}$ encodes the normalized box center coordinates, box height and width (relative to the image size). The linear projection acts as the classification branch to produce the classification results.

For the Transformer encoder in DETR, both query and key elements are of pixels in the feature maps. The inputs are of ResNet feature maps (with encoded positional embeddings). Let H and W denote the feature map height and width, respectively. The computational complexity of self-attention is of $O(H^2 W^2 C)$, which grows quadratically with the spatial size.

For the Transformer decoder in DETR, the input includes both feature maps from the encoder, and N object queries represented by learnable positional embeddings (e.g., $N = 100$). There are two types of attention modules in the decoder, namely, cross-attention and self-attention modules. In the cross-attention modules, object queries extract features from the feature maps. The query elements are of the object queries, and key elements are of the output feature maps from the encoder. In it, $N_q = N$, $N_k = H \times W$ and the complexity of the cross-attention is of $O(HWC^2 + NHWC)$. The complexity grows linearly with the spatial size of feature maps. In the self-attention modules, object queries interact with each other, so as to capture their relations. The query and key elements are both of the object queries. In it, $N_q = N_k = N$, and the complexity of the self-attention module is of $O(2NC^2 + N^2 C)$. The complexity is acceptable with moderate number of object queries.

DETR is an attractive design for object detection, which removes the need for many hand-designed components. However, it also has its own issues. These issues can be mainly attributed to the deficits of Transformer attention in handling image feature maps as key elements: (1) DETR has relatively low performance in detecting small objects. Modern object detectors use high-resolution feature maps to better detect small objects. However, high-resolution feature maps would lead to an unacceptable complexity for the self-attention module in the Transformer encoder of DETR, which has a quadratic complexity with the spatial size of input feature maps. (2) Compared with modern object detectors, DETR requires many more training epochs to converge. This is mainly because the attention modules processing image features are difficult to train. For example, at initialization, the cross-attention modules are almost of average attention on the whole feature maps. While, at the end of the training, the attention maps are learned to be very sparse, focusing only on the object

其中m表示注意力头的索引，$\boldsymbol{W}'_m \in \mathbb{R}^{C_v \times C}$和$\boldsymbol{W}_m \in \mathbb{R}^{C \times C_v}$为可学习权重（默认为$C_v = C/M$）。注意力权重$A_{mqk} \propto \exp\{\frac{\boldsymbol{z}_q^T \boldsymbol{U}_m^T \boldsymbol{V}_m \boldsymbol{x}_k}{\sqrt{C_v}}\}$通过$\sum_{k \in \Omega_k} A_{mqk} = 1$进行归一化处理，其中$\boldsymbol{U}_m, \boldsymbol{V}_m \in \mathbb{R}^{C_v \times C}$同样为可学习权重。为区分不同空间位置，表征特征$\boldsymbol{z}_q$和$\boldsymbol{x}_k$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，$\boldsymbol{U}_m \boldsymbol{z}_q$和$\boldsymbol{V}_m \boldsymbol{x}_k$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$\boldsymbol{x} \in \mathbb{R}^{C \times H \times W}$，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$\boldsymbol{b} \in [0,1]^4$，其中$\boldsymbol{b} = \{b_x, b_y, b_w, b_h\}$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$O(2NC^2 + N^2 C)$。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。这些问题主要可归因于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*}。

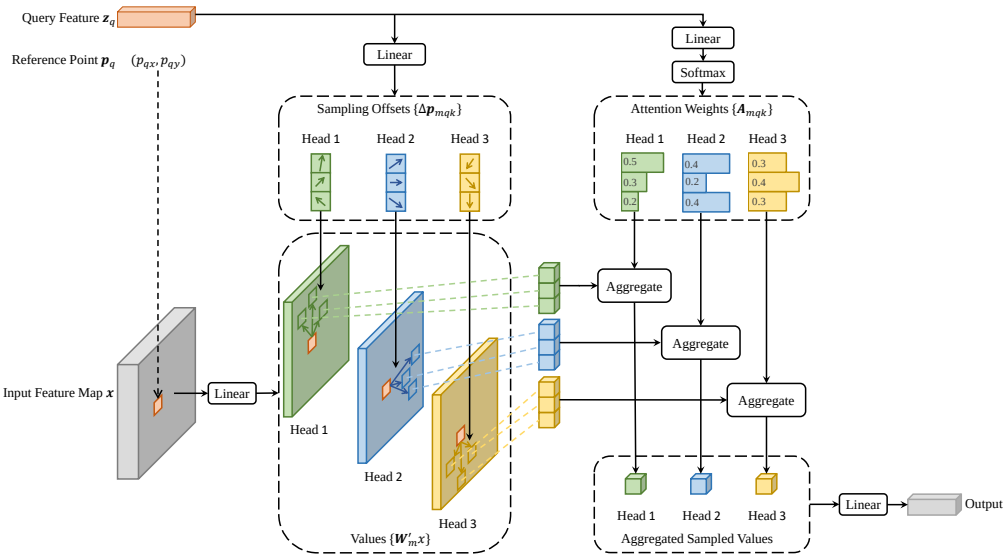


Figure 2: Illustration of the proposed deformable attention module.

extremities. It seems that DETR requires a long training schedule to learn such significant changes in the attention maps.

4 METHOD

4.1 DEFORMABLE TRANSFORMERS FOR END-TO-END OBJECT DETECTION

Deformable Attention Module. The core issue of applying Transformer attention on image feature maps is that it would look over all possible spatial locations. To address this, we present a *deformable attention module*. Inspired by deformable convolution (Dai et al., 2017; Zhu et al., 2019b), the deformable attention module only attends to a small set of key sampling points around a reference point, regardless of the spatial size of the feature maps, as shown in Fig. 2. By assigning only a small fixed number of keys for each query, the issues of convergence and feature spatial resolution can be mitigated.

Given an input feature map $\boldsymbol{x} \in \mathbb{R}^{C \times H \times W}$, let q index a query element with content feature $\boldsymbol{z}_q$ and a 2-d reference point $\boldsymbol{p}_q$, the deformable attention feature is calculated by

$$\text{DeformAttn}(\boldsymbol{z}_q, \boldsymbol{p}_q, \boldsymbol{x}) = \sum_{m=1}^{M} \boldsymbol{W}_m \Big[\sum_{k=1}^{K} A_{mqk} \cdot \boldsymbol{W}'_m \boldsymbol{x}(\boldsymbol{p}_q + \Delta \boldsymbol{p}_{mqk}) \Big], \tag{2}$$

where m indexes the attention head, k indexes the sampled keys, and K is the total sampled key number ($K \ll HW$). $\Delta \boldsymbol{p}_{mqk}$ and A_{mqk} denote the sampling offset and attention weight of the k^{th} sampling point in the m^{th} attention head, respectively. The scalar attention weight A_{mqk} lies in the range $[0, 1]$, normalized by $\sum_{k=1}^{K} A_{mqk} = 1$. $\Delta \boldsymbol{p}_{mqk} \in \mathbb{R}^2$ are of 2-d real numbers with unconstrained range. As $\boldsymbol{p}_q + \Delta \boldsymbol{p}_{mqk}$ is fractional, bilinear interpolation is applied as in Dai et al. (2017) in computing $\boldsymbol{x}(\boldsymbol{p}_q + \Delta \boldsymbol{p}_{mqk})$. Both $\Delta \boldsymbol{p}_{mqk}$ and A_{mqk} are obtained via linear projection over the query feature $\boldsymbol{z}_q$. In implementation, the query feature $\boldsymbol{z}_q$ is fed to a linear projection operator of $3MK$ channels, where the first $2MK$ channels encode the sampling offsets $\Delta \boldsymbol{p}_{mqk}$, and the remaining MK channels are fed to a $\mathrm{softmax}$ operator to obtain the attention weights A_{mqk}.

The deformable attention module is designed for processing convolutional feature maps as key elements. Let N_q be the number of query elements, when MK is relatively small, the complexity of the deformable attention module is of $O(2N_qC^2 + \min(HWC^2, N_qKC^2))$ (See Appendix A.1 for details). When it is applied in DETR encoder, where $N_q = HW$, the complexity becomes $O(HWC^2)$, which is of linear complexity with the spatial size. When it is applied as the cross-attention modules

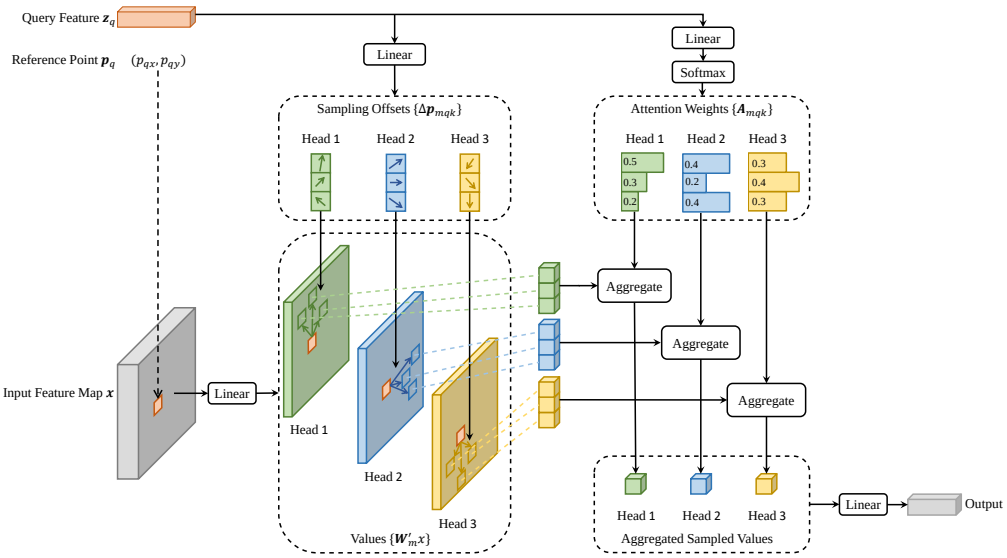


图2：所提出的可变形注意力模块示意图。

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*deformableattention 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$\boldsymbol{x} \in \mathbb{R}^{C\times H\times W}$，令$q$索引一个具有内容特征$\boldsymbol{z}_q$和二维参考点$\boldsymbol{p}_q$的查询元素，可变形注意力特征的计算方式为

$$\text{DeformAttn}(\boldsymbol{z}_q, \boldsymbol{p}_q, \boldsymbol{x}) = \sum_{m=1}^{M} \boldsymbol{W}_m \Big[\sum_{k=1}^{K} A_{mqk} \cdot \boldsymbol{W}'_m \boldsymbol{x}(\boldsymbol{p}_q + \Delta \boldsymbol{p}_{mqk})\Big], \tag{2}$$

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（$K \ll HW$）。$\Delta \boldsymbol{p}_{mqk}$和$A_{mqk}$分别代表第$m^{\text{th}}$个注意力头中第$k^{\text{th}}$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。$\Delta \boldsymbol{p}_{mqk} \in \mathbb{R}^2$为二维实数，其范围不受限制。由于$\boldsymbol{p}_q + \Delta \boldsymbol{p}_{mqk}$为分数形式，计算$\boldsymbol{x}(\boldsymbol{p}_q + \Delta \boldsymbol{p}_{mqk})$时采用了Dai等人(2017)中的双线性插值法。$\Delta \boldsymbol{p}_{mqk}$和$A_{mqk}$均通过对查询特征$\boldsymbol{z}_q$进行线性投影获得。具体实现中，查询特征$\boldsymbol{z}_q$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$\Delta \boldsymbol{p}_{mqk}$，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$O(2N_qC^2 + \min(HWC^2, N_qKC^2))$（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$O(HWC^2)$，与空间尺寸呈线性关系。当作为交叉注意力模块应用时

in DETR decoder, where $N_q = N$ (N is the number of object queries), the complexity becomes $O(NKC^2)$, which is irrelevant to the spatial size HW.

Multi-scale Deformable Attention Module. Most modern object detection frameworks benefit from multi-scale feature maps (Liu et al., 2020). Our proposed deformable attention module can be naturally extended for multi-scale feature maps.

Let $\{\boldsymbol{x}^l\}_{l=1}^L$ be the input multi-scale feature maps, where $\boldsymbol{x}^l \in \mathbb{R}^{C \times H_l \times W_l}$. Let $\hat{\boldsymbol{p}}_q \in [0,1]^2$ be the normalized coordinates of the reference point for each query element q, then the multi-scale deformable attention module is applied as

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

where m indexes the attention head, l indexes the input feature level, and k indexes the sampling point. $\Delta \boldsymbol{p}_{mlqk}$ and A_{mlqk} denote the sampling offset and attention weight of the k^{th} sampling point in the l^{th} feature level and the m^{th} attention head, respectively. The scalar attention weight A_{mlqk} is normalized by $\sum_{l=1}^{L}\sum_{k=1}^{K} A_{mlqk} = 1$. Here, we use normalized coordinates $\hat{\boldsymbol{p}}_q \in [0,1]^2$ for the clarity of scale formulation, in which the normalized coordinates $(0,0)$ and $(1,1)$ indicate the top-left and the bottom-right image corners, respectively. Function $\phi_l(\hat{\boldsymbol{p}}_q)$ in Equation 3 re-scales the normalized coordinates $\hat{\boldsymbol{p}}_q$ to the input feature map of the l-th level. The multi-scale deformable attention is very similar to the previous single-scale version, except that it samples LK points from multi-scale feature maps instead of K points from single-scale feature maps.

The proposed attention module will degenerate to deformable convolution (Dai et al., 2017), when $L = 1$, $K = 1$, and $\boldsymbol{W}'_m \in \mathbb{R}^{C_v \times C}$ is fixed as an identity matrix. Deformable convolution is designed for single-scale inputs, focusing only on one sampling point for each attention head. However, our multi-scale deformable attention looks over multiple sampling points from multi-scale inputs. The proposed (multi-scale) deformable attention module can also be perceived as an efficient variant of Transformer attention, where a pre-filtering mechanism is introduced by the deformable sampling locations. When the sampling points traverse all possible locations, the proposed attention module is equivalent to Transformer attention.

Deformable Transformer Encoder. We replace the Transformer attention modules processing feature maps in DETR with the proposed multi-scale deformable attention module. Both the input and output of the encoder are of multi-scale feature maps with the same resolutions. In encoder, we extract multi-scale feature maps $\{\boldsymbol{x}^l\}_{l=1}^{L-1}$ ($L = 4$) from the output feature maps of stages C_3 through C_5 in ResNet (He et al., 2016) (transformed by a 1×1 convolution), where C_l is of resolution 2^l lower than the input image. The lowest resolution feature map $\boldsymbol{x}^L$ is obtained via a 3×3 stride 2 convolution on the final C_5 stage, denoted as C_6. All the multi-scale feature maps are of $C = 256$ channels. Note that the top-down structure in FPN (Lin et al., 2017a) is not used, because our proposed multi-scale deformable attention in itself can exchange information among multi-scale feature maps. The constructing of multi-scale feature maps are also illustrated in Appendix A.2. Experiments in Section 5.2 show that adding FPN will not improve the performance.

In application of the multi-scale deformable attention module in encoder, the output are of multi-scale feature maps with the same resolutions as the input. Both the key and query elements are of pixels from the multi-scale feature maps. For each query pixel, the reference point is itself. To identify which feature level each query pixel lies in, we add a scale-level embedding, denoted as $\boldsymbol{e}_l$, to the feature representation, in addition to the positional embedding. Different from the positional embedding with fixed encodings, the scale-level embedding $\{\boldsymbol{e}_l\}_{l=1}^L$ are randomly initialized and jointly trained with the network.

Deformable Transformer Decoder. There are cross-attention and self-attention modules in the decoder. The query elements for both types of attention modules are of object queries. In the cross-attention modules, object queries extract features from the feature maps, where the key elements are of the output feature maps from the encoder. In the self-attention modules, object queries interact with each other, where the key elements are of the object queries. Since our proposed deformable attention module is designed for processing convolutional feature maps as key elements, we only replace each cross-attention module to be the multi-scale deformable attention module, while leaving the self-attention modules unchanged. For each object query, the 2-d normalized coordinate of the

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$\{\boldsymbol{x}^l\}_{l=1}^L$为输入的多尺度特征图，其中$\boldsymbol{x}^l \in \mathbb{R}^{C \times H_l \times W_l}$。令$\hat{\boldsymbol{p}}_q \in [0,1]^2$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

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

其中m索引注意力头，l索引输入特征层级，k索引采样点。$\Delta \boldsymbol{p}_{mlqk}$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$\hat{\boldsymbol{p}}_q$ ∈[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$\phi_l(\hat{\boldsymbol{p}}_q)$将归一化坐标$\hat{\boldsymbol{p}}_q$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$\boldsymbol{W}'_m \in \mathbb{R}^{C_v \times C}$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1×1卷积转换）提取多尺度特征图$\{\boldsymbol{x}^l\}_{l=1}^{L-1}$ (L =至)，其中C_l的分辨率比输入图像低2^l倍。最低分辨率特征图$\boldsymbol{x}^L$是通过在最后的C_5阶段应用3×3步长2卷积获得，记为C_6。所有多尺度特征图均具有C =个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$\boldsymbol{e}_l$。与固定编码的位置嵌入不同，尺度层级嵌入$\{\boldsymbol{e}_l\}_{l=1}^L$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*}

reference point $\hat{\boldsymbol{p}}_q$ is predicted from its object query embedding via a learnable linear projection followed by a $\mathrm{sigmoid}$ function.

Because the multi-scale deformable attention module extracts image features around the reference point, we let the detection head predict the bounding box as relative offsets w.r.t. the reference point to further reduce the optimization difficulty. The reference point is used as the initial guess of the box center. The detection head predicts the relative offsets w.r.t. the reference point. Check Appendix A.3 for the details. In this way, the learned decoder attention will have strong correlation with the predicted bounding boxes, which also accelerates the training convergence.

By replacing Transformer attention modules with deformable attention modules in DETR, we establish an efficient and fast converging detection system, dubbed as Deformable DETR (see Fig. 1).

4.2 Additional Improvements and Variants for Deformable DETR

Deformable DETR opens up possibilities for us to exploit various variants of end-to-end object detectors, thanks to its fast convergence, and computational and memory efficiency. Due to limited space, we only introduce the core ideas of these improvements and variants here. The implementation details are given in Appendix A.4.

Iterative Bounding Box Refinement. This is inspired by the iterative refinement developed in optical flow estimation (Teed & Deng, 2020). We establish a simple and effective iterative bounding box refinement mechanism to improve detection performance. Here, each decoder layer refines the bounding boxes based on the predictions from the previous layer.

Two-Stage Deformable DETR. In the original DETR, object queries in the decoder are irrelevant to the current image. Inspired by two-stage object detectors, we explore a variant of Deformable DETR for generating region proposals as the first stage. The generated region proposals will be fed into the decoder as object queries for further refinement, forming a two-stage Deformable DETR.

In the first stage, to achieve high-recall proposals, each pixel in the multi-scale feature maps would serve as an object query. However, directly setting object queries as pixels will bring unacceptable computational and memory cost for the self-attention modules in the decoder, whose complexity grows quadratically with the number of queries. To avoid this problem, we remove the decoder and form an encoder-only Deformable DETR for region proposal generation. In it, each pixel is assigned as an object query, which directly predicts a bounding box. Top scoring bounding boxes are picked as region proposals. No NMS is applied before feeding the region proposals to the second stage.

5 Experiment

Dataset. We conduct experiments on COCO 2017 dataset (Lin et al., 2014). Our models are trained on the train set, and evaluated on the val set and test-dev set.

Implementation Details. ImageNet (Deng et al., 2009) pre-trained ResNet-50 (He et al., 2016) is utilized as the backbone for ablations. Multi-scale feature maps are extracted without FPN (Lin et al., 2017a). $M = 8$ and $K = 4$ are set for deformable attentions by default. Parameters of the deformable Transformer encoder are shared among different feature levels. Other hyper-parameter setting and training strategy mainly follow DETR (Carion et al., 2020), except that Focal Loss (Lin et al., 2017b) with loss weight of 2 is used for bounding box classification, and the number of object queries is increased from 100 to 300. We also report the performance of DETR-DC5 with these modifications for a fair comparison, denoted as DETR-DC5$^+$. By default, models are trained for 50 epochs and the learning rate is decayed at the 40-th epoch by a factor of 0.1. Following DETR(Carion et al., 2020), we train our models using Adam optimizer (Kingma & Ba, 2015) with base learning rate of 2×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay of 10^{-4}. Learning rates of the linear projections, used for predicting object query reference points and sampling offsets, are multiplied by a factor of 0.1. Run time is evaluated on NVIDIA Tesla V100 GPU.

5.1 Comparison with DETR

As shown in Table 1, compared with Faster R-CNN + FPN, DETR requires many more training epochs to converge, and delivers lower performance at detecting small objects. Compared with

参考点$\hat{\boldsymbol{p}}_q$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$^+$。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2×10^{-4}、β_1 =0.9、β_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, Deformable DETR achieves better performance (especially on small objects) with 10× less training epochs. Detailed convergence curves are shown in Fig. 3. With the aid of iterative bounding box refinement and two-stage paradigm, our method can further improve the detection accuracy.

Our proposed Deformable DETR has on par FLOPs with Faster R-CNN + FPN and DETR-DC5. But the runtime speed is much faster (1.6×) than DETR-DC5, and is just 25% slower than Faster R-CNN + FPN. The speed issue of DETR-DC5 is mainly due to the large amount of memory access in Transformer attention. Our proposed deformable attention can mitigate this issue, at the cost of unordered memory access. Thus, it is still slightly slower than traditional convolution.

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

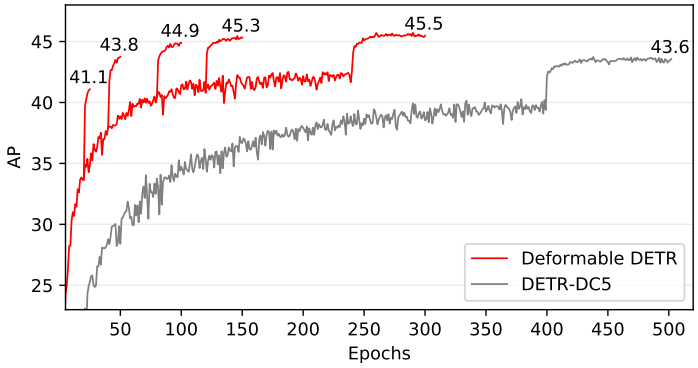


Figure 3: Convergence curves of Deformable DETR and DETR-DC5 on COCO 2017 val set. For Deformable DETR, we explore different training schedules by varying the epochs at which the learning rate is reduced (where the AP score leaps).

5.2 Ablation Study on Deformable Attention

Table 2 presents ablations for various design choices of the proposed deformable attention module. Using multi-scale inputs instead of single-scale inputs can effectively improve detection accuracy with 1.7% AP, especially on small objects with 2.9% AP_S. Increasing the number of sampling points K can further improve 0.9% AP. Using multi-scale deformable attention, which allows information exchange among different scale levels, can bring additional 1.5% improvement in AP. Because the cross-level feature exchange is already adopted, adding FPNs will not improve the performance. When multi-scale attention is not applied, and $K = 1$, our (multi-scale) deformable attention module degenerates to deformable convolution, delivering noticeable lower accuracy.

5.3 Comparison with State-of-the-art Methods

Table 3 compares the proposed method with other state-of-the-art methods. Iterative bounding box refinement and two-stage mechanism are both utilized by our models in Table 3. With ResNet-101 and ResNeXt-101 (Xie et al., 2017), our method achieves 48.7 AP and 49.0 AP without bells and whistles, respectively. By using ResNeXt-101 with DCN (Zhu et al., 2019b), the accuracy rises to 50.1 AP. With additional test-time augmentations, the proposed method achieves 52.3 AP.

DETR和Deformable DETR以少10×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×），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+表示采用Focal Loss和300个目标查询的DETR-DC5。

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

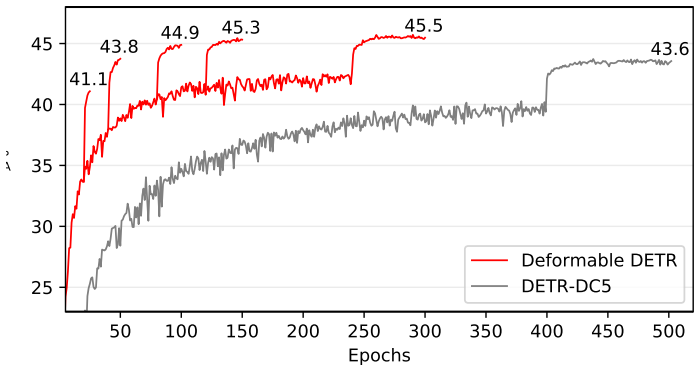


图3：Deformable DETR与DETR-DC5在COCO 2017验证集上的收敛曲线。对于Deformable DETR，我们通过改变学习率下降的周期（即AP分数跃升的节点）来探索不同的训练方案。

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_S。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。

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	w/o	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	w/o	43.8	62.6	47.7	26.4	47.1	58.0

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

Deformable DETR is an end-to-end object detector, which is efficient and fast-converging. It enables us to explore more interesting and practical variants of end-to-end object detectors. At the core of Deformable DETR are the (multi-scale) deformable attention modules, which is an efficient attention mechanism in processing image feature maps. We hope our work opens up new possibilities in exploring end-to-end object detection.

ACKNOWLEDGMENTS

The work is supported by the National Key R&D Program of China (2020AAA0105200), Beijing Academy of Artificial Intelligence, and the National Natural Science Foundation of China under grand No.U19B2044 and No.61836011.

REFERENCES

Joshua Ainslie, Santiago Ontanon, Chris Alberti, Philip Pham, Anirudh Ravula, and Sumit Sanghai. Etc: Encoding long and structured data in transformers. *arXiv preprint arXiv:2004.08483*, 2020.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Jared Davis, Tamas Sarlos, David Belanger, Lucy Colwell, and Adrian Weller. Masked language modeling for proteins via linearly scalable long-context transformers. *arXiv preprint arXiv:2006.03555*, 2020.

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_{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	w/o	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	w/o	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_{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 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 Likhosherstov、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。

Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *CVPR*, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019.

Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. In *ICCV*, 2019.

Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *ICCV*, 2019.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. *arXiv preprint arXiv:2006.16236*, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *ICLR*, 2020.

Tao Kong, Fuchun Sun, Chuanqi Tan, Huaping Liu, and Wenbing Huang. Deep feature pyramid reconfiguration for object detection. In *ECCV*, 2018.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017a.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017b.

Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *IJCV*, 2020.

Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. In *ICLR*, 2018a.

Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *CVPR*, 2018b.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *ICML*, 2018.

Jiezhong Qiu, Hao Ma, Omer Levy, Scott Wen-tau Yih, Sinong Wang, and Jie Tang. Blockwise self-attention for long document understanding. *arXiv preprint arXiv:1911.02972*, 2019.

Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Stand-alone self-attention in vision models. In *NeurIPS*, 2019.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. Efficient content-based sparse attention with routing transformers. *arXiv preprint arXiv:2003.05997*, 2020.

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 François 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。

林倞毅（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。

Guanglu Song, Yu Liu, and Xiaogang Wang. Revisiting the sibling head in object detector. In *CVPR*, 2020.

Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *CVPR*, 2020.

Yi Tay, Dara Bahri, Liu Yang, Donald Metzler, and Da-Cheng Juan. Sparse sinkhorn attention. In *ICML*, 2020a.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *arXiv preprint arXiv:2009.06732*, 2020b.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020.

Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. *arXiv preprint arXiv:2003.07853*, 2020a.

Sinong Wang, Belinda Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020b.

Felix Wu, Angela Fan, Alexei Baevski, Yann N Dauphin, and Michael Auli. Pay less attention with lightweight and dynamic convolutions. In *ICLR*, 2019.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

Hang Xu, Lewei Yao, Wei Zhang, Xiaodan Liang, and Zhenguo Li. Auto-fpn: Automatic network architecture adaptation for object detection beyond classification. In *ICCV*, 2019.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *arXiv preprint arXiv:2007.14062*, 2020.

Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *CVPR*, 2020.

Qijie Zhao, Tao Sheng, Yongtao Wang, Zhi Tang, Ying Chen, Ling Cai, and Haibin Ling. M2det: A single-shot object detector based on multi-level feature pyramid network. In *AAAI*, 2019.

Xizhou Zhu, Dazhi Cheng, Zheng Zhang, Stephen Lin, and Jifeng Dai. An empirical study of spatial attention mechanisms in deep networks. In *ICCV*, 2019a.

Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *CVPR*, 2019b.

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（ ukasz Kaiser）和伊利亚·波洛苏金（Illia Polosukhin）。注意力机制就是你所需要的一切。发表于*NeurIPS*，2017年。

王慧宇、朱利安·施泰因·Bradley Green、Hartwig Adam、Alan Yuille与陈良杰。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。Linformer：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。

A APPENDIX

A.1 COMPLEXITY FOR DEFORMABLE ATTENTION

Supposes the number of query elements is N_q, in the deformable attention module (see Equation 2), the complexity for calculating the sampling coordinate offsets $\Delta \boldsymbol{p}_{mqk}$ and attention weights A_{mqk} is of $O(3N_qCMK)$. Given the sampling coordinate offsets and attention weights, the complexity of computing Equation 2 is $O(N_qC^2 + N_qKC^2 + 5N_qKC)$, where the factor of 5 in $5N_qKC$ is because of bilinear interpolation and the weighted sum in attention. On the other hand, we can also calculate $\boldsymbol{W}'_m \boldsymbol{x}$ before sampling, as it is independent to query, and the complexity of computing Equation 2 will become as $O(N_qC^2 + HWC^2 + 5N_qKC)$. So the overall complexity of deformable attention is $O(N_qC^2 + \min(HWC^2, N_qKC^2) + 5N_qKC + 3N_qCMK)$. In our experiments, $M = 8$, $K \leq 4$ and $C = 256$ by default, thus $5K + 3MK < C$ and the complexity is of $O(2N_qC^2 + \min(HWC^2, N_qKC^2))$.

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

As discussed in Section 4.1 and illustrated in Figure 4, the input multi-scale feature maps of the encoder $\{\boldsymbol{x}^l\}_{l=1}^{L-1}$ ($L = 4$) are extracted from the output feature maps of stages C_3 through C_5 in ResNet (He et al., 2016) (transformed by a 1×1 convolution). The lowest resolution feature map $\boldsymbol{x}^L$ is obtained via a 3×3 stride 2 convolution on the final C_5 stage. Note that FPN (Lin et al., 2017a) is not used, because our proposed multi-scale deformable attention in itself can exchange information among multi-scale feature maps.

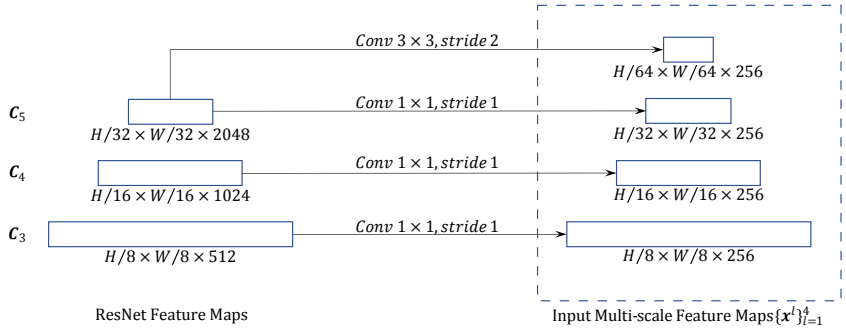


Figure 4: Constructing mult-scale feature maps for Deformable DETR.

A.3 BOUNDING BOX PREDICTION IN DEFORMABLE DETR

Since the multi-scale deformable attention module extracts image features around the reference point, we design the detection head to predict the bounding box as relative offsets w.r.t. the reference point to further reduce the optimization difficulty. The reference point is used as the initial guess of the box center. The detection head predicts the relative offsets w.r.t. the reference point $\hat{\boldsymbol{p}}_q = (\hat{p}_{qx}, \hat{p}_{qy})$, i.e., $\hat{\boldsymbol{b}}_q = \{\sigma\big(b_{qx} + \sigma^{-1}(\hat{p}_{qx})\big), \sigma\big(b_{qy} + \sigma^{-1}(\hat{p}_{qy})\big), \sigma(b_{qw}), \sigma(b_{qh})\}$, where $b_{q\{x,y,w,h\}} \in \mathbb{R}$ are predicted by the detection head. σ and σ^{-1} denote the sigmoid and the inverse sigmoid function, respectively. The usage of σ and σ^{-1} is to ensure $\hat{\boldsymbol{b}}$ is of normalized coordinates, as $\hat{\boldsymbol{b}}_q \in [0, 1]^4$. In this way, the learned decoder attention will have strong correlation with the predicted bounding boxes, which also accelerates the training convergence.

A.4 MORE IMPLEMENTATION DETAILS

Iterative Bounding Box Refinement. Here, each decoder layer refines the bounding boxes based on the predictions from the previous layer. Suppose there are D number of decoder layers (e.g., $D = 6$), given a normalized bounding box $\hat{\boldsymbol{b}}_q^{d-1}$ predicted by the $(d-1)$-th decoder layer, the d-th

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$\Delta \boldsymbol{p}_{mqk}$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后，计算公式2的复杂度为$O(N_qC^2 + N_qKC^2 + 5N_qKC)$，其中$5N_qKC$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$\boldsymbol{W}'_m\boldsymbol{x}$，因为它与查询无关，此时计算公式2的复杂度将变为$O(N_qC^2 + HWC^2 + 5N_qKC)$。因此，可变形注意力的总体复杂度为$O(N_qC^2 + \min(HWC^2, N_qKC^2) + 5N_qKC + 3N_qCMK)$。在我们的实验中，默认设置$M = 8$、$K \leq 4$和$C = 256$，故$5K + 3MK < C$，复杂度为$O(2N_qC^2 + \min(HWC^2, N_qKC^2))$。

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$\{\boldsymbol{x}^l\}_{l=1}^{L-1}$ (L =至)提取自ResNet（He等人，2016）中C_3至C_5阶段的输出特征图（通过1×1卷积变换得到）。最低分辨率特征图$\boldsymbol{x}^L$是通过在最终C_5阶段应用3×3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。

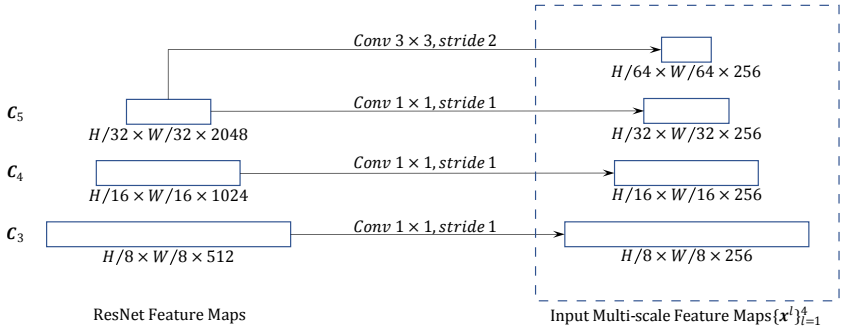


图4：为可变形DETR构建多尺度特征图。

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$\hat{\boldsymbol{p}}_q = (\hat{p}_{qx}, \hat{p}_{qy})$，即$\hat{\boldsymbol{b}}_q = \{\sigma\big(b_{qx} + \sigma^{-1}(\hat{p}_{qx})\big), \sigma\big(b_{qy} + \sigma^{-1}(\hat{p}_{qy})\big), \sigma(b_{qw}), \sigma(b_{qh})\}$，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σ和σ^{-1}的目的是确保$\hat{\boldsymbol{b}}$为归一化坐标，因为$\hat{\boldsymbol{b}}_q \in [0,1]^4$。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。

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$\hat{\boldsymbol{b}}_q^{d-1}$，第$d$层

decoder layer refines the box as

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

where $d \in \{1, 2, ..., D\}$, $\Delta b_{q\{x,y,w,h\}}^d \in \mathbb{R}$ are predicted at the d-th decoder layer. Prediction heads for different decoder layers do not share parameters. The initial box is set as $\hat{b}_{qx}^0 = \hat{p}_{qx}$, $\hat{b}_{qy}^0 = \hat{p}_{qy}$, $\hat{b}_{qw}^0 = 0.1$, and $\hat{b}_{qh}^0 = 0.1$. The system is robust to the choice of b_{qw}^0 and b_{qh}^0. We tried setting them as 0.05, 0.1, 0.2, 0.5, and achieved similar performance. To stabilize training, similar to Teed & Deng (2020), the gradients only back propagate through $\Delta b_{q\{x,y,w,h\}}^d$, and are blocked at $\sigma^{-1}(\hat{b}_{q\{x,y,w,h\}}^{d-1})$.

In iterative bounding box refinement, for the d-th decoder layer, we sample key elements respective to the box $\hat{\boldsymbol{b}}_q^{d-1}$ predicted from the $(d-1)$-th decoder layer. For Equation 3 in the cross-attention module of the d-th decoder layer, $(\hat{b}_{qx}^{d-1}, \hat{b}_{qy}^{d-1})$ serves as the new reference point. The sampling offset $\Delta \boldsymbol{p}_{mlqk}$ is also modulated by the box size, as $(\Delta p_{mlqkx}\, \hat{b}_{qw}^{d-1}, \Delta p_{mlqky}\, \hat{b}_{qh}^{d-1})$. Such modifications make the sampling locations related to the center and size of previously predicted boxes.

Two-Stage Deformable DETR. In the first stage, given the output feature maps of the encoder, a detection head is applied to each pixel. The detection head is of a 3-layer FFN for bounding box regression, and a linear projection for bounding box binary classification (i.e., foreground and background), respectively. Let i index a pixel from feature level $l_i \in \{1, 2, ..., L\}$ with 2-d normalized coordinates $\hat{\boldsymbol{p}}_i = (\hat{p}_{ix}, \hat{p}_{iy}) \in [0, 1]^2$, its corresponding bounding box is predicted by

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

where the base object scale s is set as 0.05, $\Delta b_{i\{x,y,w,h\}} \in \mathbb{R}$ are predicted by the bounding box regression branch. The Hungarian loss in DETR is used for training the detection head.

Given the predicted bounding boxes in the first stage, top scoring bounding boxes are picked as region proposals. In the second stage, these region proposals are fed into the decoder as initial boxes for the *iterative bounding box refinement*, where the positional embeddings of object queries are set as positional embeddings of region proposal coordinates.

Initialization for Multi-scale Deformable Attention. In our experiments, the number of attention heads is set as $M = 8$. In multi-scale deformable attention modules, $\boldsymbol{W}_m' \in \mathbb{R}^{C_v \times C}$ and $\boldsymbol{W}_m \in \mathbb{R}^{C \times C_v}$ are randomly initialized. Weight parameters of the linear projection for predicting A_{mlqk} and $\Delta \boldsymbol{p}_{mlqk}$ are initialized to zero. Bias parameters of the linear projection are initialized to make $A_{mlqk} = \frac{1}{LK}$ and $\{\Delta \boldsymbol{p}_{1lqk} = (-k, -k), \Delta \boldsymbol{p}_{2lqk} = (-k, 0), \Delta \boldsymbol{p}_{3lqk} = (-k, k), \Delta \boldsymbol{p}_{4lqk} = (0, -k), \Delta \boldsymbol{p}_{5lqk} = (0, k), \Delta \boldsymbol{p}_{6lqk} = (k, -k), \Delta \boldsymbol{p}_{7lqk} = (k, 0), \Delta \boldsymbol{p}_{8lqk} = (k, k)\}$ $(k \in \{1, 2, ...K\})$ at initialization.

For *iterative bounding box refinement*, the initialized bias parameters for $\Delta \boldsymbol{p}_{mlqk}$ prediction in the decoder are further multiplied with $\frac{1}{2K}$, so that all the sampling points at initialization are within the corresponding bounding boxes predicted from the previous decoder layer.

A.5 What Deformable DETR looks at?

For studying what Deformable DETR looks at to give final detection result, we draw the gradient norm of each item in final prediction (i.e., x/y coordinate of object center, width/height of object bounding box, category score of this object) with respect to each pixel in the image, as shown in Fig. 5. According to Taylor's theorem, the gradient norm can reflect how much the output would be changed relative to the perturbation of the pixel, thus it could show us which pixels the model mainly relys on for predicting each item.

The visualization indicates that Deformable DETR looks at extreme points of the object to determine its bounding box, which is similar to the observation in DETR (Carion et al., 2020). More concretely, Deformable DETR attends to left/right boundary of the object for x coordinate and width, and top/bottom boundary for y coordinate and height. Meanwhile, different to DETR (Carion et al., 2020), our Deformable DETR also looks at pixels inside the object for predicting its category.

解码器层将框细化为

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

其中$d \in \{1,2,...,D\}$、$\Delta b_{q\{x,y,w,h\}}^d \in \mathbb{R}$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$\hat{\boldsymbol{b}}_q^{d-1}$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$\Delta \boldsymbol{p}_{mlqk}$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$l_i \in \{1,2,...,L\}$的像素索引，其二维归一化坐标为$\hat{\boldsymbol{p}}_i = (\hat{p}_{ix}, \hat{p}_{iy}) \in [0,1]^2$，其对应边界框由以下公式预测得出：

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

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，$\boldsymbol{W}_m' \in \mathbb{R}^{C_v \times C}$和$\boldsymbol{W}_m \in \mathbb{R}^{C \times C_v}$被随机初始化。用于预测$A_{mlqk}$和$\Delta \boldsymbol{p}_{mlqk}$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$\{\Delta \boldsymbol{p}_{1lqk} = (-k,-k), \Delta \boldsymbol{p}_{2lqk} = (-k,$在初始状态下为0$), \Delta \boldsymbol{p}_{3lqk} = (-k,k), \Delta \boldsymbol{p}_{4lqk} = (0, -k), \Delta \boldsymbol{p}_{5lqk} = (0,k), \Delta \boldsymbol{p}_{6lqk} = (k,-k), \Delta \boldsymbol{p}_{7lqk} = (k,0), \Delta \boldsymbol{p}_{8lqk} = (k,k)\}$ $(k \in \{1,2,...K\})$。

对于*iterative bounding box refinement*，解码器中用于$\Delta \boldsymbol{p}_{mlqk}$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。

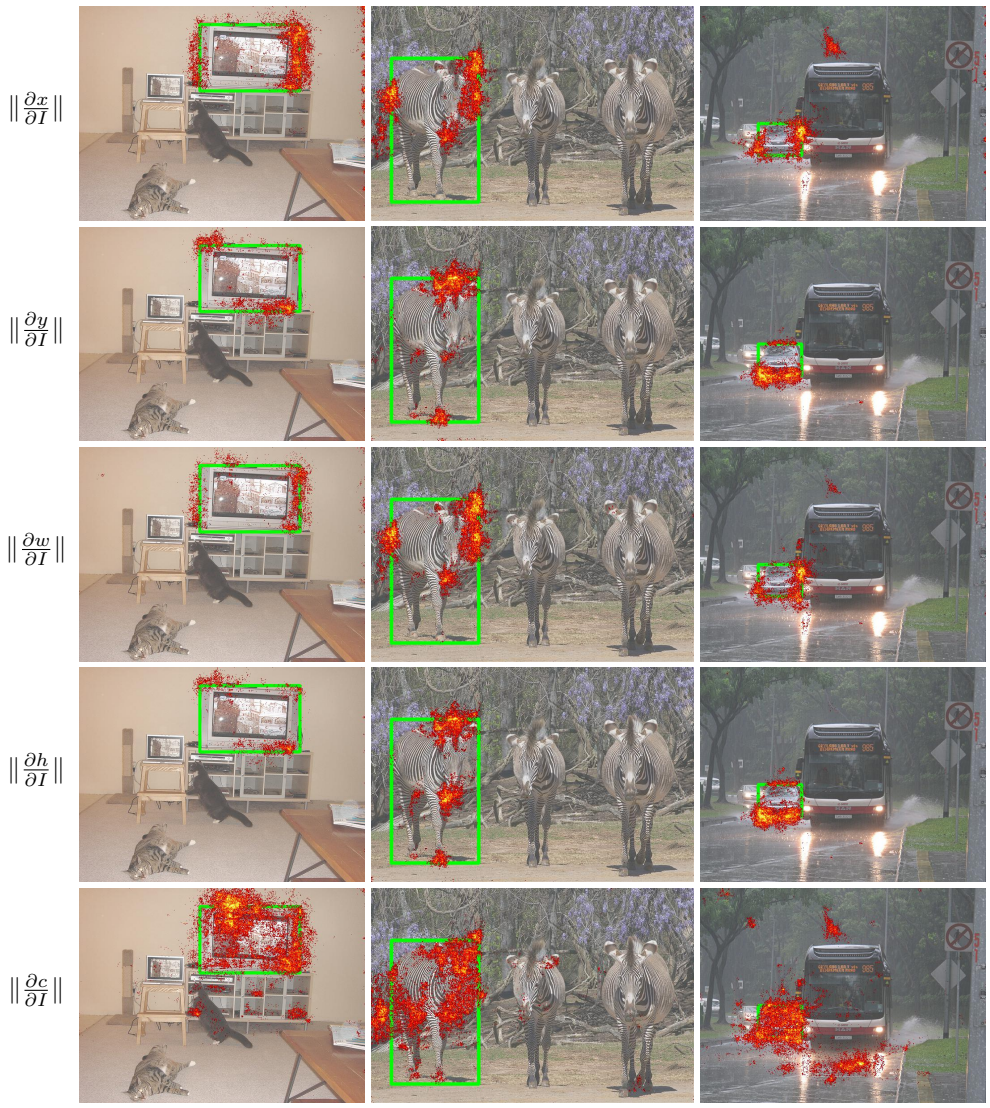


Figure 5: The gradient norm of each item (coordinate of object center (x, y), width/height of object bounding box w/h, category score c of this object) in final detection result with respect to each pixel in input image I.

A.6 Visualization of Multi-scale Deformable Attention

For better understanding learned multi-scale deformable attention modules, we visualize sampling points and attention weights of the last layer in encoder and decoder, as shown in Fig. 6. For readibility, we combine the sampling points and attention weights from feature maps of different resolutions into one picture.

Similar to DETR (Carion et al., 2020), the instances are already separated in the encoder of Deformable DETR. While in the decoder, our model is focused on the whole foreground instance instead of only extreme points as observed in DETR (Carion et al., 2020). Combined with the visualization of $\|\frac{\partial c}{\partial I}\|$ in Fig. 5, we can guess the reason is that our Deformable DETR needs not only extreme points but also interior points to detemine object category. The visualization also demonstrates that the proposed multi-scale deformable attention module can adapt its sampling points and attention weights according to different scales and shapes of the foreground object.

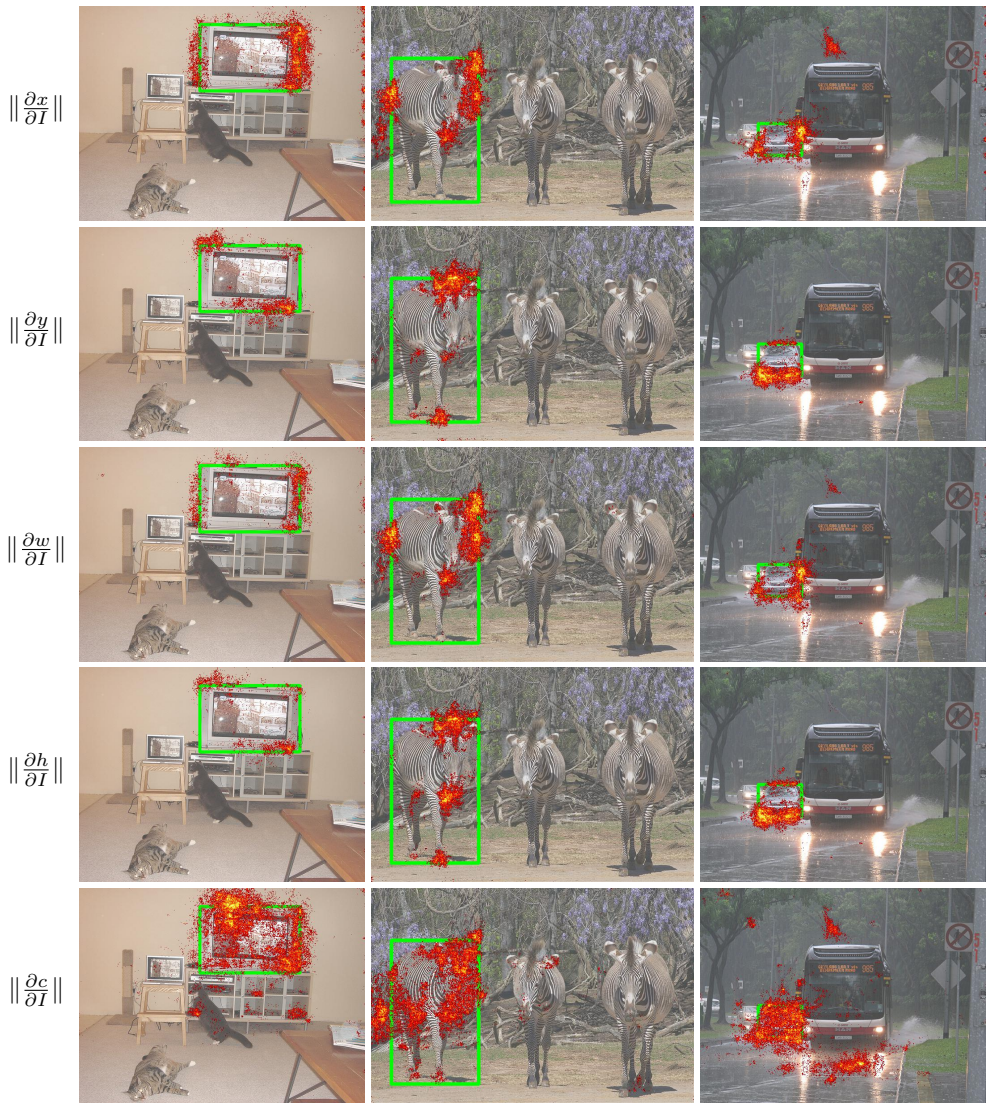
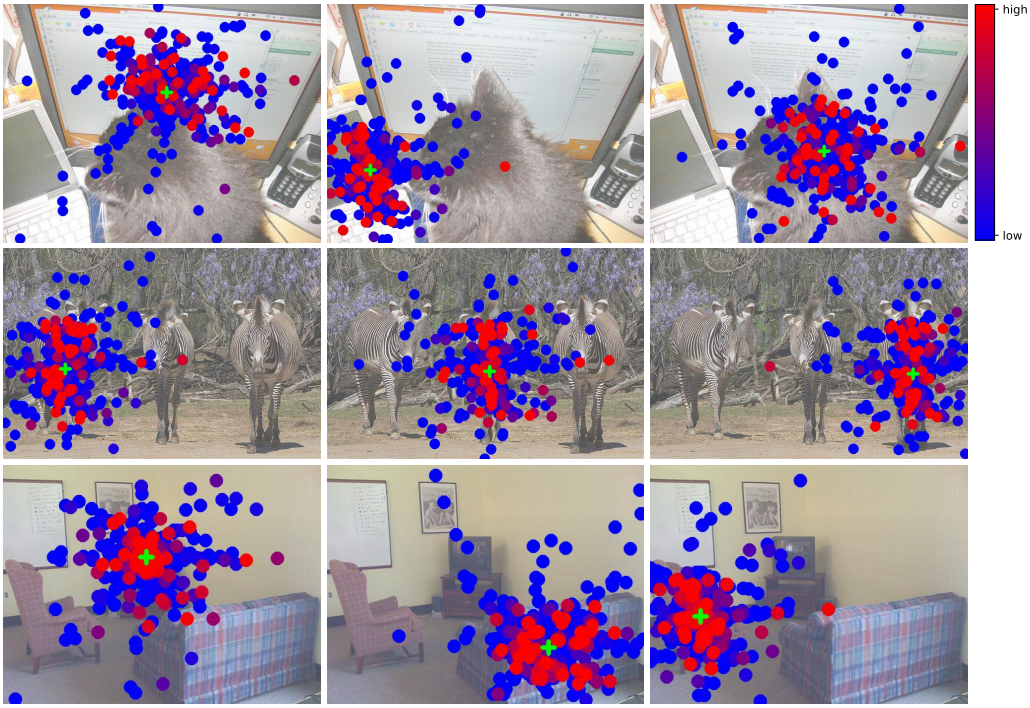


图5：最终检测结果中各项（物体中心坐标x, y、物体边界框宽度/高度w/h、该物体类别得分c）相对于输入图像I各像素的梯度范数。

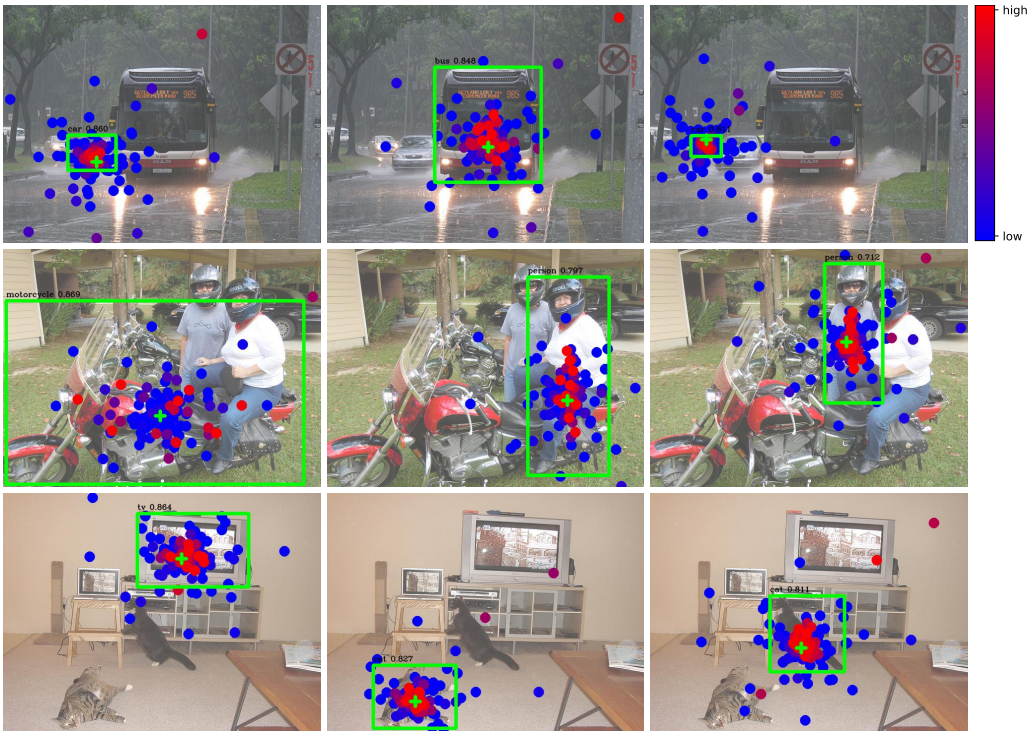
A.6 多尺度可变形注意力的可视化

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。

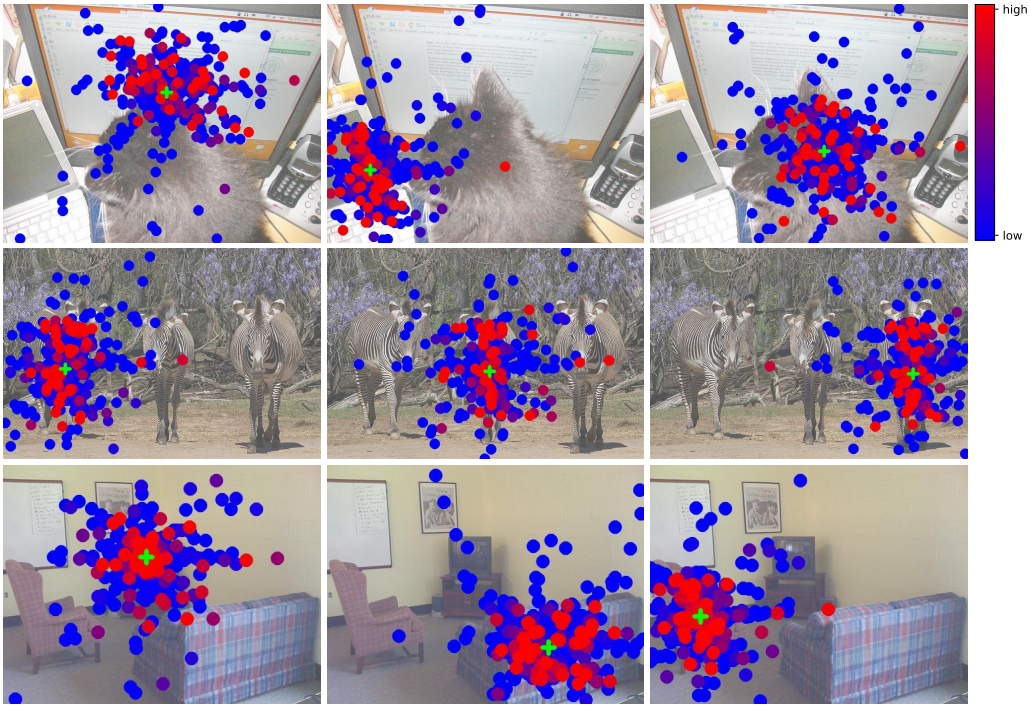


(a) multi-scale deformable self-attention in encoder

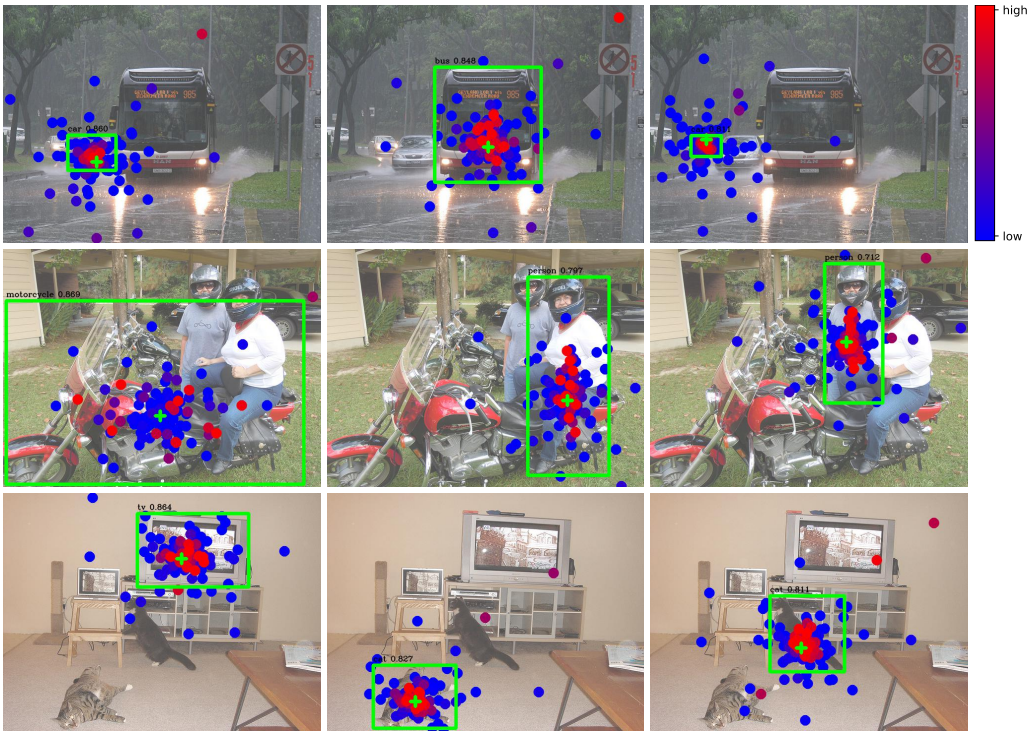


(b) multi-scale deformable cross-attention in decoder

Figure 6: Visualization of multi-scale deformable attention. For readibility, we draw the sampling points and attention weights from feature maps of different resolutions in one picture. Each sampling point is marked as a filled circle whose color indicates its correspoinding attention weight. The reference point is shown as green cross marker, which is also equivalent to query point in encoder. In decoder, the predicted bounding box is shown as a green rectangle and the category and confidence score are texted just above it.



(a) multi-scale deformable self-attention in encoder



(解码器中的)多尺度可变形交叉注意力

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 NOTATIONS

Table 4: Lookup table for notations in the paper.

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
$\boldsymbol{z}_q$	input feature of q^{th} query
$\boldsymbol{p}_q$	2-d coordinate of reference point for q^{th} query
$\hat{\boldsymbol{p}}_q$	normalized 2-d coordinate of reference point for q^{th} query
$\boldsymbol{x}$	input feature map (input feature of key elements)
$\boldsymbol{x}_k$	input feature of k^{th} key
$\boldsymbol{x}^l$	input feature map of l^{th} feature level
$\Delta \boldsymbol{p}_{mqk}$	sampling offset of q^{th} query to k^{th} key at m^{th} head
$\Delta \boldsymbol{p}_{mlqk}$	sampling offset of q^{th} query to k^{th} key in l^{th} feature level at m^{th} head
$\boldsymbol{W}_m$	output projection matrix at m^{th} head
$\boldsymbol{U}_m$	input query projection matrix at m^{th} head
$\boldsymbol{V}_m$	input key projection matrix at m^{th} head
$\boldsymbol{W}'_m$	input value projection matrix at m^{th} head
$\phi_l(\hat{\boldsymbol{p}})$	unnormalized 2-d coordinate of $\hat{\boldsymbol{p}}$ in l^{th} feature level
$\exp$	exponential function
σ	sigmoid function
σ^{-1}	inverse sigmoid function

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
$\boldsymbol{z}_q$	input feature of q^{th} query
$\boldsymbol{p}_q$	2-d coordinate of reference point for q^{th} query
$\hat{\boldsymbol{p}}_q$	normalized 2-d coordinate of reference point for q^{th} query
$\boldsymbol{x}$	input feature map (input feature of key elements)
$\boldsymbol{x}_k$	input feature of k^{th} key
$\boldsymbol{x}^l$	input feature map of l^{th} feature level
$\Delta\boldsymbol{p}_{mqk}$	sampling offset of q^{th} query to k^{th} key at m^{th} head
$\Delta\boldsymbol{p}_{mlqk}$	sampling offset of q^{th} query to k^{th} key in l^{th} feature level at m^{th} head
$\boldsymbol{W}_m$	output projection matrix at m^{th} head
$\boldsymbol{U}_m$	input query projection matrix at m^{th} head
$\boldsymbol{V}_m$	input key projection matrix at m^{th} head
$\boldsymbol{W}'_m$	input value projection matrix at m^{th} head
$\phi_l(\hat{\boldsymbol{p}})$	unnormalized 2-d coordinate of $\hat{\boldsymbol{p}}$ in l^{th} feature level
$\exp$	exponential function
σ	sigmoid function
σ^{-1}	inverse sigmoid function