Conditional DETR for Fast Training Convergence

Depu Meng[1*] Xiaokang Chen[2*] Zejia Fan[2] Gang Zeng[2]
Houqiang Li[1] Yuhui Yuan[3] Lei Sun[3] Jingdong Wang[3†]
[1]University of Science and Technology of China [2]Peking University [3]Microsoft Research Asia

Abstract

The recently-developed DETR approach applies the transformer encoder and decoder architecture to object detection and achieves promising performance. In this paper, we handle the critical issue, slow training convergence, and present a conditional cross-attention mechanism for fast DETR training. Our approach is motivated by that the cross-attention in DETR relies highly on the content embeddings for localizing the four extremities and predicting the box, which increases the need for high-quality content embeddings and thus the training difficulty.

Our approach, named conditional DETR, learns a conditional spatial query from the decoder embedding for decoder multi-head cross-attention. The benefit is that through the conditional spatial query, each cross-attention head is able to attend to a band containing a distinct region, e.g., one object extremity or a region inside the object box. This narrows down the spatial range for localizing the distinct regions for object classification and box regression, thus relaxing the dependence on the content embeddings and easing the training. Empirical results show that conditional DETR converges $6.7\times$ *faster for the backbones R*50 *and R*101 *and* $10\times$ *faster for stronger backbones DC*5*-R*50 *and DC*5*-R*101*. Code is available at* https://github.com/Atten4Vis/ConditionalDETR.

1. Introduction

The DEtection TRansformer (DETR) method [3] applies the transformer encoder and decoder architecture to object detection and achieves good performance. It effectively eliminates the need for many hand-crafted components, including non-maximum suppression and anchor generation.

The DETR approach suffers from slow convergence on training, and needs 500 training epochs to get good performance. The very recent work, deformable DETR [53], handles this issue by replacing the global dense attention (self-attention and cross-attention) with deformable attention that attends to a small set of key sampling points and using the high-resolution and multi-scale encoder. Instead, we still use the global dense attention and propose an improved decoder cross-attention mechanism for accelerating the training process.

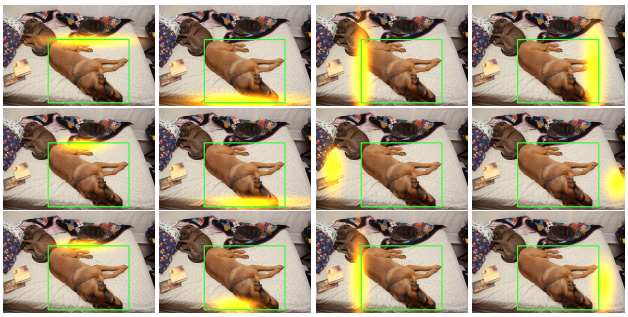

Figure 1. Comparison of spatial attention weight maps for our conditional DETR-R50 with 50 training epochs (the first row), the original DETR-R50 with 50 training epochs (the second row), and the original DETR-R50 with 500 training epochs (the third row). The maps for our conditional DETR and DETR trained with 500 epochs are able to highlight the four extremity regions satisfactorily. In contrast, the spatial attention weight maps responsible for the left and right edges (the third and fourth images in the second row) from DETR trained with 50 epochs cannot highlight the extremities satisfactorily. The green box is the ground-truth box.

Our approach is motivated by high dependence on content embeddings and minor contributions made by the spatial embeddings in cross-attention. The empirical results in DETR [3] show that if removing the positional embeddings in keys and the object queries from the second decoder layer and only using the content embeddings in keys and queries, the detection AP drops slightly[1].

Figure 1 (the second row) shows that the spatial attention weight maps from the cross-attention in DETR trained with 50 epochs. One can see that two among the four maps do not correctly highlight the bands for the corresponding

*The two authors share first authorship, and the order was determined by rolling dice. This work was done when D. Meng, X. Chen, and Z. Fan were interns at Microsoft Research, Beijing, P.R. China

†Corresponding author.

[1]The minor AP drop 1.4 is reported on R50 with 300 epochs in Table 3 from [3]. We empirically got the consistent observation: the AP drops to 34.0 from 34.9 for 50 training epochs.

条件式DETR实现快速训练收敛

孟德璞[1*] 陈小康[2*] 范泽佳[2] 曾刚[2] 李厚强[1] 袁钰慧[3] 孙磊[3] 王井东[3†] [1]中国科学技术大学 [2]北京大学 [3]微软亚洲研究院

摘要

The recently-developed DETR approach applies the transformer encoder and decoder architecture to object detection and achieves promising performance. In this paper, we handle the critical issue, slow training convergence, and present a conditional cross-attention mechanism for fast DETR training. Our approach is motivated by that the cross-attention in DETR relies highly on the content embeddings for localizing the four extremities and predicting the box, which increases the need for high-quality content embeddings and thus the training difficulty.

Our approach, named conditional DETR, learns a conditional spatial query from the decoder embedding for decoder multi-head cross-attention. The benefit is that through the conditional spatial query, each cross-attention head is able to attend to a band containing a distinct region, e.g., one object extremity or a region inside the object box. This narrows down the spatial range for localizing the distinct regions for object classification and box regression, thus relaxing the dependence on the content embeddings and easing the training. Empirical results show that conditional DETR converges 6.7× *faster for the backbones R*50 *and R*101 *and* 10 × *faster for stronger backbones DC*5-*R*50 *and DC*5-*R*101 *. Code is available at* https://github.com/Atten4Vis/ConditionalDETR.

1. 引言

DETR（DEtection TRansformer）方法[3]将Transformer编码器-解码器架构应用于目标检测任务，并取得了优异的性能表现。该方法有效摒弃了包括非极大值抑制和锚框生成在内的诸多手工设计组件。

DETR方法在训练过程中收敛速度较慢，需要500个训练周期才能获得良好的性能。最近的工作，可变形DETR[53],

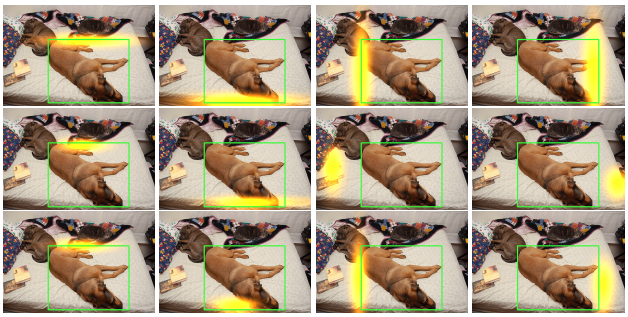

图1. 我们提出的条件式DETR-R50（50训练周期，第一行）、原始DETR-R50（50训练周期，第二行）与原始DETR-R50（500训练周期，第三行）的空间注意力权重图对比。我们的条件式DETR和经过500周期训练的DETR生成的权重图能较好地突出四个肢体末端区域。相比之下，仅训练50周期的DETR生成的左右边缘空间注意力权重图（第二行第三、四幅图像）无法充分突出肢体末端。绿色框标注的是真实标注框。

通过采用可变形注意力机制替代全局密集注意力（自注意力和交叉注意力），仅关注少量关键采样点，并利用高分辨率和多尺度编码器来处理这一问题。而我们则仍保留全局密集注意力，并提出一种改进的解码器交叉注意力机制，以加速训练过程。

我们的方法源于对内容嵌入的高度依赖以及空间嵌入在交叉注意力中贡献较小。DETR[3]中的实证结果表明，若从第二解码器层移除键中的位置嵌入和对象查询，仅使用键和查询中的内容嵌入，检测AP会略有下降[1]。

图1（第二行）展示了经过50个epoch训练的DETR中交叉注意力生成的空间注意力权重图。可以看出，四张图中有两张未能正确突出对应波段的{v*}。

*The two authors share first authorship, and the order was determined by rolling dice. This work was done when D. Meng, X. Chen, and Z. Fan were interns at Microsoft Research, Beijing, P.R. China

†Corresponding author.

[1]The minor AP drop 1.4 is reported on R50 with 300 epochs in Table 3 from [3]. We empirically got the consistent observation: the AP drops to 34.0 from 34.9 for 50 training epochs.

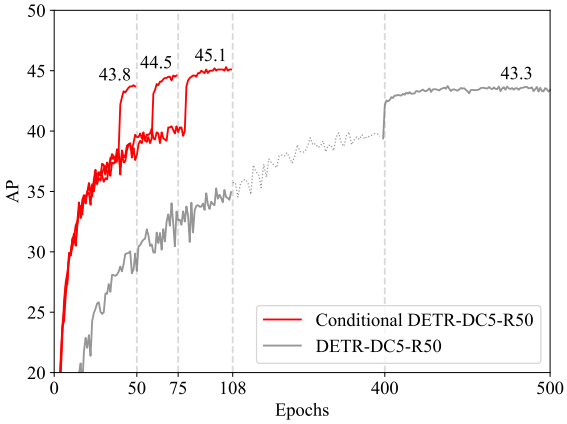

Figure 2. Convergence curves for conditional DETR-DC5-R50 and DETR-DC5-R50 on COCO 2017 `val`. The conditional DETR is trained for 50, 75, 108 epochs. Conditional DETR training is converged much faster than DETR.

extremities, thus weak at shrinking the spatial range for the content queries to precisely localize the extremities. The reasons are that (i) the spatial queries, i.e., object queries, only give the general attention weight map without exploiting the specific image information; and that (ii) due to short training the content queries are not strong enough to match the spatial keys well as they are also used to match the content keys. This increases the dependence on high-quality content embeddings, thus increasing the training difficulty.

We present a conditional DETR approach, which learns a conditional spatial embedding for each query from the corresponding previous decoder output embedding, to form a so-called conditional spatial query for decoder multi-head cross-attention. The conditional spatial query is predicted by mapping the information for regressing the object box to the embedding space, the same to the space that the 2D coordinates of the keys are also mapped to.

We empirically observe that using the spatial queries and keys, each cross-attention head spatially attends to a band containing the object extremity or a region inside the object box (Figure 1, the first row). This shrinks the spatial range for the content queries to localize the effective regions for class and box prediction. As a result, the dependence on the content embeddings is relaxed and the training is easier. The experiments show that conditional DETR converges $6.7\times$ faster for the backbones R50 and R101 and $10\times$ faster for stronger backbones DC5-R50 and DC5-R101. Figure 2 gives the convergence curves for conditional DETR and the original DETR [3].

2. Related Work

Anchor-based and anchor-free detection. Most existing object detection approaches make predictions from initial guesses that are carefully designed. There are two main initial guesses: anchor boxes or object centers. The anchor box-based methods inherit the ideas from the proposal-based method, Fast R-CNN. Example methods include Faster R-CNN [9], SSD [26], YOLOv2 [31], YOLOv3 [32], YOLOv4 [1], RetinaNet [24], Cascade R-CNN [2], Libra R-CNN [29], TSD [35] and so on.

The anchor-free detectors predict the boxes at points near the object centers. Typical methods include YOLOv1 [30], CornerNet [21], ExtremeNet [50], CenterNet [49, 6], FCOS [39] and others [23, 28, 52, 19, 51, 22, 15, 46, 47].

DETR and its variants. DETR successfully applies transformers to object detection, effectively removing the need for many hand-designed components like non-maximum suppression or initial guess generation. The high computation complexity issue, caused by the global encoder self-attention, is handled in adaptive clustering transformer [48] and by sparse attentions in deformable DETR [53].

The other critical issue, slow training convergence, has been attracting a lot of recent research attention. The TSP (transformer-based set prediction) approach [37] eliminates the cross-attention modules and combines the FCOS and R-CNN-like detection heads. Deformable DETR [53] adopts deformable attention, which attends to sparse positions learned from the content embedding, to replace decoder cross-attention.

The spatially modulated co-attention (SMCA) approach [7], which is concurrent to our approach, is very close to our approach. It modulates the DETR multi-head global cross-attentions with Gaussian maps around a few (shifted) centers that are learned from the decoder embeddings, to focus more on a few regions inside the estimated box. In contrast, the proposed conditional DETR approach learns the conditional spatial queries from the decoder content embeddings, and predicts the spatial attention weight maps without human-crafting the attention attenuation, which highlight four extremities for box regression, and distinct regions inside the object for classification.

Conditional and dynamic convolution. The proposed conditional spatial query scheme is related to conditional convolutional kernel generation. Dynamic filter network [16] learns the convolutional kernels from the input, which is applied to instance segmentation in CondInst [38] and SOLOv2 [42] for learning instance-dependent convolutional kernels. CondConv [44] and dynamic convolution [4] mix convolutional kernels with the weights learned from the input. SENet [14], GENet [13] abd Lite-HRNet [45] learn from the input the channel-wise weights.

These methods learn from the input the convolutional kernel weights and then apply the convolutions to the input. In contrast, the linear projection in our approach is learned from the decoder embeddings for representing the displacement and scaling information.

Transformers. The transformer [40] relies on the at-

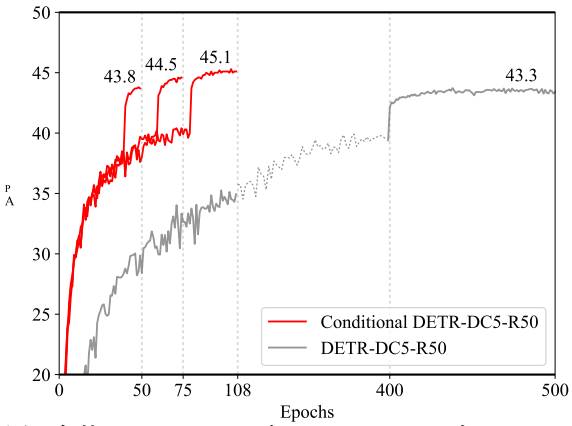

图2. 条件DETR-DC5-R50与DETR-DC5-R50在COCO 2017验证集上的收敛曲线。条件DETR分别训练了50、75、108个周期。条件DETR的训练收敛速度明显快于DETR。

末端区域，因此在缩小内容查询的空间范围以精确定位末端时表现较弱。原因在于：(i) 空间查询（即对象查询）仅提供一般的注意力权重图，未能充分利用特定图像信息；(ii) 由于训练时间较短，内容查询在匹配空间键时不够强健，因为它们还需同时匹配内容键。这加大了对高质量内容嵌入的依赖，从而增加了训练难度。

我们提出了一种条件式DETR方法，该方法从解码器前一层的输出嵌入中为每个查询学习一个条件空间嵌入，从而构建所谓的条件空间查询，用于解码器的多头交叉注意力机制。条件空间查询通过将回归目标框的信息映射至嵌入空间来预测，该空间与键的二维坐标所映射的空间相同。公式标记{v*}保持不变。

我们通过实证观察发现，使用空间查询和键时，每个交叉注意力头在空间上会关注包含物体 extremities（极值点）的带状区域或物体框内部的某个区域（图1第一行）。这缩小了内容查询的空间范围，使其能够更精准地定位用于类别和边界框预测的有效区域。因此，模型对内容嵌入的依赖得以减轻，训练过程更为容易。实验表明，对于骨干网络R50和R101，条件式DETR的收敛速度提升了6.7×倍；而对于更强的骨干网络DC5-R50和DC5-R101，收敛速度更是加快了10×倍。图2展示了条件式DETR与原始DETR[3]的收敛曲线对比。

2. 相关工作

基于锚点与无锚点的检测。现有的大多数目标检测方法都是从精心设计的初始猜测出发进行预测。主要的初始猜测有两种：锚框或物体中心。这些方...

基于锚框的方法继承了基于提议的方法Fast R-CNN的思想。典型方法包括Faster R-CNN [9]、SSD [26]、YOLOv2 [31]、YOLOv3 [32]、YOLOv4 [1]、RetinaNet [24]、Cascade R-CNN [2]、Libra R-CNN [29]、TSD [35]等。

无锚框检测器在物体中心附近的点预测边界框。典型方法包括YOLOv1 [30]、CornerNet [21]、ExtremeNet [50]、CenterNet [49, 6]、FCOS [39]及其他[23, 28, 52, 19, 51, 22, 15, 46, 47]。

DETR及其变体。DETR成功将transformer架构应用于目标检测领域，有效消除了对非极大值抑制或初始猜测生成等众多人工设计组件的需求。针对全局编码器自注意力机制导致的高计算复杂度问题，自适应聚类transformer[48]和可变形DETR[53]中的稀疏注意力机制分别提出了解决方案。

另一个关键问题——训练收敛速度慢，近期吸引了大量研究关注。基于Transformer的集合预测（TSP）方法[37]取消了交叉注意力模块，融合了FCOS与类R-CNN检测头。可变形DETR[53]采用可变形注意力机制，该机制通过学习内容嵌入生成的稀疏位置进行注意力计算，以此替代解码器中的交叉注意力模块。

空间调制协同注意力（SMCA）方法[7]与我们提出的方法同期问世，且极为相似。该方法通过在解码器嵌入学习到的若干（偏移）中心周围施加高斯映射，对DETR多头全局交叉注意力进行调制，从而更集中于预测框内的特定区域。相比之下，本文提出的条件式DETR方法从解码器内容嵌入中学习条件空间查询，无需人工设计注意力衰减机制即可预测空间注意力权重图——这些权重图会突出显示用于边界框回归的四个极值点，以及用于分类的目标内部不同区域。

条件与动态卷积。所提出的条件空间查询方案与条件卷积核生成相关。动态滤波器网络[16]从输入中学习卷积核，这一方法被应用于CondInst[38]和SOLOv2[42]中的实例分割任务，以学习实例相关的卷积核。CondConv[44]和动态卷积[4]则将卷积核与从输入学习得到的权重进行混合。SENet[14]、GENet[13]以及Lite-HRNet[45]则从输入中学习通道级别的权重。

这些方法从输入中学习卷积核权重，然后将卷积应用于输入。相比之下，我们方法中的线性投影是从解码器嵌入中学习而来，用于表示位移和缩放信息。

变压器。变压器[40]依赖于这一

tention mechanism, self-attention and cross-attention, to draw global dependencies between the input and the output. There are several works closely related to our approach. Gaussian transformer [11] and T-GSA (Transformer with Gaussian-weighted self-attention) [18], followed by SMCA [7], attenuate the attention weights according to the distance between target and context symbols with learned or human-crafted Gaussian variance. Similar to ours, TUPE [17] computes the attention weight also from the spatial attention weight and the content attention weight. Instead, our approach mainly focuses on the attention attenuation mechanism in a learnable form other than a Gaussian function, and potentially benefits speech enhancement [18] and natural language inference [11].

3. Conditional DETR

3.1. Overview

Pipeline. The proposed approach follows detection transformer (DETR), an end-to-end object detector, and predicts all the objects at once without the need for NMS or anchor generation. The architecture consists of a CNN backbone, a transformer encoder, a transformer decoder, and object class and box position predictors. The transformer encoder aims to improve the content embeddings output from the CNN backbone. It is a stack of multiple encoder layers, where each layer mainly consists of a self-attention layer and a feed-forward layer.

The transformer decoder is a stack of decoder layers. Each decoder layer, illustrated in Figure 3, is composed of three main layers: (1) a self-attention layer for removing duplication prediction, which performs interactions between the embeddings, outputted from the previous decoder layer and used for class and box prediction, (2) a cross-attention layer, which aggregates the embeddings output from the encoder to refine the decoder embeddings for improving class and box prediction, and (3) a feed-forward layer.

Box regression. A candidate box is predicted from each decoder embedding as follows,

$$\mathbf{b} = \text{sigmoid}(\text{FFN}(\mathbf{f}) + [\mathbf{s}^\top \; 0 \; 0]^\top). \tag{1}$$

Here, $\mathbf{f}$ is the decoder embedding. $\mathbf{b}$ is a four-dimensional vector $[b_{cx} \; b_{cy} \; b_w \; b_h]^\top$, consisting of the box center, the box width and the box height. $\text{sigmoid}()$ is used to normalize the prediction $\mathbf{b}$ to the range $[0, 1]$. $\text{FFN}()$ aims to predict the unnormalized box. $\mathbf{s}$ is the unnormalized 2D coordinate of the reference point, and is $(0, 0)$ in the original DETR. In our approach, we consider two choices: learn the reference point $\mathbf{s}$ as a parameter for each candidate box prediction, or generate it from the corresponding object query.

Category prediction. The classification score for each candidate box is also predicted from the decoder embedding through an FNN, $\mathbf{e} = \text{FFN}(\mathbf{f})$.

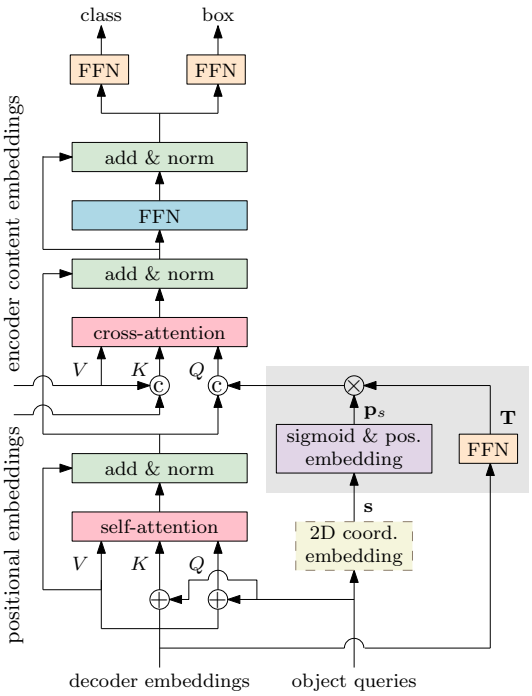

Figure 3. Illustrating one decoder layer in conditional DETR. The main difference from the original DETR [3] lies in the input queries and the input keys for cross-attention. The conditional spatial query is predicted from learnable 2D coordinates $\mathbf{s}$ and the embeddings output from the previous decoder layer, through the operations depicted in the gray-shaded box. The 2D coordinate $\mathbf{s}$ can be predicted from the object query (the dashed box), or simply learned as model parameters The spatial query (key) and the content query (key) are concatenated as the query (key). The resulting cross-attention is called conditional cross-attention. Same as DETR [3], the decoder layer is repeated 6 times.

Main work. The cross-attention mechanism aims to *localize the distinct regions, four extremities for box detection and regions inside the box for object classification, and aggregates the corresponding embeddings*. We propose a conditional cross-attention mechanism with introducing conditional spatial queries for improving the localization capability and accelerating the training process.

3.2. DETR Decoder Cross-Attention

The DETR decoder cross-attention mechanism takes three inputs: queries, keys and values. Each key is formed by adding a content key $\mathbf{c}_k$ (the content embedding output from the encoder) and a spatial key $\mathbf{p}_k$ (the positional embedding of the corresponding normalized 2D coordinate). The value is formed from the content embedding, same with the content key, output from the encoder.

In the original DETR approach, each query is formed by adding a content query $\mathbf{c}_q$ (the embedding output from the decoder self-attention), and a spatial query $\mathbf{p}_q$ (i.e., the object query $\mathbf{o}_q$). In our implementation, there are $N = 300$

注意力机制，包括自注意力与交叉注意力，用于捕捉输入与输出之间的全局依赖关系。有多项研究与我们的方法密切相关。高斯Transformer[11]和T-GSA（采用高斯加权自注意力的Transformer）[18]，以及后续的SMCA[7]，通过学习的或人工设计的高斯方差，根据目标符号与上下文符号之间的距离来衰减注意力权重。与我们的方法类似，TUPE[17]同样从空间注意力权重和内容注意力权重计算注意力权重。不同的是，我们的方法主要关注于以可学习形式而非高斯函数实现的注意力衰减机制，并可能有益于语音增强[18]和自然语言推理[11]任务。

3. 条件式DETR

3.1. 概述

流程。所提出的方法遵循检测变换器（DETR），这是一种端到端的目标检测器，能够一次性预测所有对象，无需非极大值抑制（NMS）或锚框生成。该架构由CNN主干网络、变换器编码器、变换器解码器以及目标类别与边界框位置预测器组成。变换器编码器旨在优化CNN主干网络输出的内容嵌入表示，它由多个编码层堆叠而成，每层主要包含自注意力层和前馈层。

Transformer解码器由多个解码层堆叠而成。如图3所示，每个解码层包含三个核心部分：(1) 自注意力层，用于消除重复预测，该层对来自前一解码层的嵌入表示进行交互处理，这些嵌入用于类别和边界框预测；(2) 交叉注意力层，通过聚合编码器输出的嵌入表示来优化解码器嵌入，从而提升类别和边界框预测精度；(3) 前馈神经网络层。

框回归。每个解码器嵌入会预测出一个候选框，具体方式如下，

$$\mathbf{b} = \text{sigmoid}(\text{FFN}(\mathbf{f}) + [\mathbf{s}^\top \; 0 \; 0]^\top). \tag{1}$$

这里，$\mathbf{f}$是解码器的嵌入向量。$\mathbf{b}$是一个四维向量$[b_{cx}\; b_{cy}\; b_w\; b_h]^\top$，包含框的中心点坐标、宽度和高度。sigmoid()函数用于将预测值$\mathbf{b}$归一化到[0,1]的范围内。FFN()的目标是预测未归一化的边界框。$\mathbf{s}$表示参考点的未归一化二维坐标，在原版DETR中其值为(0,0)。在我们的方法中，我们探讨了两种选择：将参考点$\mathbf{s}$作为每个候选框预测的可学习参数，或从对应的对象查询中生成该参考点。

类别预测。每个候选框的分类分数同样通过解码器嵌入经由一个FNN预测得出，{v*} FFN({v*})。

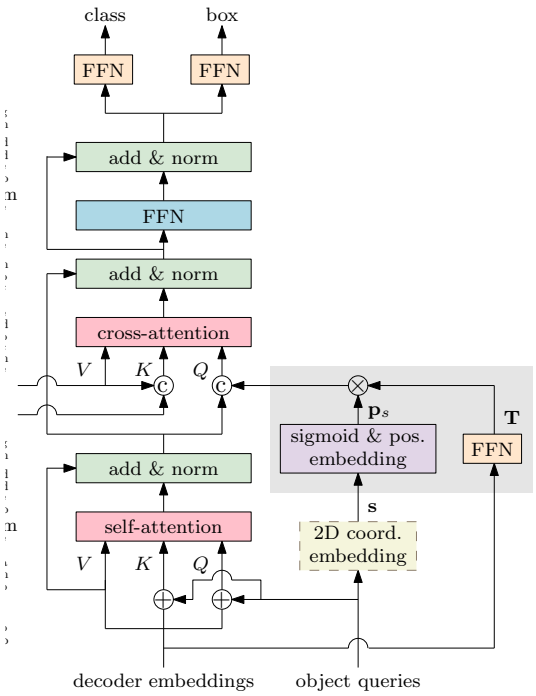

图3. 展示了条件DETR中的一个解码器层。与原始DETR[3]的主要区别在于输入查询和交叉注意力机制的输入键。条件空间查询通过灰色阴影框内描绘的操作，从可学习的二维坐标$\mathbf{s}$及前一解码器层输出的嵌入中预测得出。二维坐标$\mathbf{s}$可从对象查询（虚线框）预测，或直接作为模型参数学习。空间查询（键）与内容查询（键）拼接后形成查询（键）。由此产生的交叉注意力称为条件交叉注意力。与DETR[3]相同，解码器层重复6次。

主要工作。交叉注意力机制旨在*localize the distinct regions, four extremities for box detection and regions inside the box for object classification, and aggregates the corresponding embeddings*。我们提出了一种条件化交叉注意力机制，通过引入条件空间查询来提升定位能力并加速训练过程。

3.2. DETR解码器交叉注意力

DETR解码器的交叉注意力机制接收三个输入：查询项、键项和值项。每个键项由内容键$\mathbf{c}_k$（（编码器输出的内容嵌入））与空间键$\mathbf{p}_k$（（对应归一化二维坐标的位置嵌入））相加构成。而值项则源自编码器输出的内容嵌入，与内容键相同。

在原始的DETR方法中，每个查询由内容查询$\mathbf{c}_q$（（解码器自注意力输出的嵌入向量））和空间查询$\mathbf{p}_q$（（即对象查询$\mathbf{o}_q$））相加构成。在我们的实现中，共有N =300个

object queries, and accordingly there are N queries[2], each query outputting a candidate detect result in one decoder layer.

The attention weight is based on the dot-product between the query and the key, used for attention weight computation,

$$\begin{aligned} & (\mathbf{c}_q + \mathbf{p}_q)^\top (\mathbf{c}_k + \mathbf{p}_k) \\ = \; & \mathbf{c}_q^\top \mathbf{c}_k + \mathbf{c}_q^\top \mathbf{p}_k + \mathbf{p}_q^\top \mathbf{c}_k + \mathbf{p}_q^\top \mathbf{p}_k \\ = \; & \mathbf{c}_q^\top \mathbf{c}_k + \mathbf{c}_q^\top \mathbf{p}_k + \mathbf{o}_q^\top \mathbf{c}_k + \mathbf{o}_q^\top \mathbf{p}_k. \end{aligned} \tag{2}$$

3.3. Conditional Cross-Attention

The proposed conditional cross-attention mechanism forms the query by concatenating the content query $\mathbf{c}_q$, outputting from decoder self-attention, and the spatial query $\mathbf{p}_q$. Accordingly, the key is formed as the concatenation of the content key $\mathbf{c}_k$ and the spatial key $\mathbf{p}_k$.

The cross-attention weights consist of two components, content attention weight and spatial attention weight. The two weights are from two dot-products, content and spatial dot-products,

$$\mathbf{c}_q^\top \mathbf{c}_k + \mathbf{p}_q^\top \mathbf{p}_k. \tag{3}$$

Different from the original DETR cross-attention, our mechanism separates the roles of content and spatial queries so that spatial and content queries focus on the spatial and content attention weights, respectively.

An additional important task is to compute the spatial query $\mathbf{p}_q$ from the embedding $\mathbf{f}$ of the previous decoder layer. We first identify that the spatial information of the distinct regions are determined by the two factors together, decoder embedding and reference point. We then show how to map them to the embedding space, forming the query $\mathbf{p}_q$, so that the spatial query lies in the same space the 2D coordinates of the keys are mapped to.

The decoder embedding contains the displacements of the distinct regions with respect to the reference point. The box prediction process in Equation 1 consists of two steps: (1) predicting the box with respect to the reference point in the unnormalized space, and (2) normalizing the predicted box to the range $[0, 1]$[3].

Step (1) means that the decoder embedding $\mathbf{f}$ contains the displacements of the four extremities (forming the box) with respect to the reference point $\mathbf{s}$ in the unnormalized space. This implies that both the embedding $\mathbf{f}$ and the reference point $\mathbf{s}$ are necessary to determine the spatial information of the distinct regions, the four extremities as well as the region for predicting the classification score.

[2]For description simplicity and clearness, we drop the query, key, and value indices.

[3]The origin $(0, 0)$ in the unnormalized space for the original DETR method is mapped to $(0.5, 0.5)$ (the center in the image space) in the normalized space through the sigmoid function.

Conditional spatial query prediction. We predict the conditional spatial query from the embedding $\mathbf{f}$ and the reference point $\mathbf{s}$,

$$(\mathbf{s}, \mathbf{f}) \rightarrow \mathbf{p}_q, \tag{4}$$

so that it is aligned with the positional space which the normalized 2D coordinates of the keys are mapped to. The process is illustrated in the gray-shaded box area of Figure 3.

We normalize the reference point $\mathbf{s}$ and then map it to a 256-dimensional sinusoidal positional embedding in the same way as the positional embedding for keys:

$$\mathbf{p}_s = \text{sinusoidal}(\text{sigmoid}(\mathbf{s})). \tag{5}$$

We then map the displacement information contained in the decoder embedding $\mathbf{f}$ to a linear projection in the same space through an FFN consisting of learnable linear projection + ReLU + learnable linear projection: $\mathbf{T} = \text{FFN}(\mathbf{f})$.

The conditional spatial query is computed by transforming the reference point in the embedding space: $\mathbf{p}_q = \mathbf{T}\mathbf{p}_s$. We choose the simple and computationally-efficient projection matrix, a diagonal matrix. The 256 diagonal elements are denoted as a vector $\boldsymbol{\lambda}_q$. The conditional spatial query is computed by the element-wise multiplication:

$$\mathbf{p}_q = \mathbf{T}\mathbf{p}_s = \boldsymbol{\lambda}_q \odot \mathbf{p}_s. \tag{6}$$

Multi-head cross-attention. Following DETR [3], we adopt the standard multi-head cross-attention mechanism. Object detection usually needs to implicitly or explicitly localize the four object extremities for accurate box regression and localize the object region for accurate object classification. The multi-head mechanism is beneficial to disentangle the localization tasks.

We perform multi-head parallel attentions by projecting the queries, the keys, and the values $M = 8$ times with learned linear projections to low dimensions. The spatial and content queries (keys) are separately projected to each head with different linear projections. The projections for values are the same as the original DETR and are only for the contents.

3.4. Visualization and Analysis

Visualization. Figure 4 visualizes the attention weight maps for each head: the spatial attention weight maps, the content attention weight maps, and the combined attention weight maps. The maps are soft-max normalized over the spatial dot-products $\mathbf{p}_q^\top \mathbf{p}_k$, the content dot-products $\mathbf{c}_q^\top \mathbf{c}_k$, and the combined dot-products $\mathbf{c}_q^\top \mathbf{c}_k + \mathbf{p}_q^\top \mathbf{p}_k$. We show 5 out of the 8 maps, and other three are the duplicates, corresponding to bottom and top extremities, and a small region inside the object box[4].

[4]The duplicates might be different for models trained several times, but the detection performance is almost the same.

目标查询，相应地有N个查询[2]，每个查询在一个解码器层中输出一个候选检测结果。

注意力权重基于查询与键之间的点积，用于注意力权重的计算，

$$\begin{aligned}&(\mathbf{c}_q+\mathbf{p}_q)^\top(\mathbf{c}_k+\mathbf{p}_k)\\ =\ &\mathbf{c}_q^\top\mathbf{c}_k+\mathbf{c}_q^\top\mathbf{p}_k+\mathbf{p}_q^\top\mathbf{c}_k+\mathbf{p}_q^\top\mathbf{p}_k\\ =\ &\mathbf{c}_q^\top\mathbf{c}_k+\mathbf{c}_q^\top\mathbf{p}_k+\mathbf{o}_q^\top\mathbf{c}_k+\mathbf{o}_q^\top\mathbf{p}_k.\end{aligned}\tag{2}$$

3.3. 条件交叉注意力

提出的条件交叉注意力机制通过将内容查询$\mathbf{c}_q$（来自解码器自注意力的输出）与空间查询$\mathbf{p}_q$拼接起来形成查询向量。相应地，键向量则由内容键$\mathbf{c}_k$和空间键$\mathbf{p}_k$的拼接构成。

交叉注意力权重由两部分组成：内容注意力权重和空间注意力权重。这两部分权重分别来源于两个点积运算——内容点积和空间点积。

$$\mathbf{c}_q^\top\mathbf{c}_k+\mathbf{p}_q^\top\mathbf{p}_k.\tag{3}$$

不同于原始DETR的交叉注意力机制，我们的方法将内容查询与空间查询的角色分离，使得空间查询和内容查询分别专注于空间注意力权重和内容注意力权重。

另一项重要任务是从前一解码器层的嵌入**f**中计算出空间查询$\mathbf{p}_q$。我们首先明确，不同区域的空间信息由解码器嵌入和参考点这两个因素共同决定。接着，我们展示了如何将它们映射到嵌入空间，形成查询$\mathbf{p}_q$，从而使空间查询与键的二维坐标所映射到的空间保持一致。

解码器嵌入包含了各独立区域相对于参考点的位移。公式1中的框预测过程包含两个步骤：(1) 在非归一化空间中预测相对于参考点的框，(2) 将预测框归一化至[0, 1][3]范围内。

步骤（1）意味着解码器嵌入**f**包含了四个极端点（构成方框）相对于未归一化空间中参考点**s**的位移。这表明，要确定不同区域的空间信息——包括四个极端点以及用于预测分类得分的区域——既需要嵌入**f**，也需要参考点**s**。

[2]For description simplicity and clearness, we drop the query, key, and value indices.

[3]The origin $(0, 0)$ in the unnormalized space for the original DETR method is mapped to $(0.5, 0.5)$ (the center in the image space) in the normalized space through the $\mathrm{sigmoid}$ function.

条件空间查询预测。我们从嵌入**f**和参考点**s**预测条件空间查询，

$$(\mathbf{s},\mathbf{f})\rightarrow\mathbf{p}_q,\tag{4}$$

使其与归一化后的2D关键点坐标所映射到的位置空间对齐。该过程在图3的灰色阴影框区域中进行了说明。

我们将参考点**s**归一化，然后以与键的位置嵌入相同的方式，将其映射到一个256维的正弦位置嵌入中：

$$\mathbf{p}_s=\mathrm{sinusoidal}(\mathrm{sigmoid}(\mathbf{s})).\tag{5}$$

随后，我们通过一个由可学习线性投影+、ReLU激活+及可学习线性投影$\mathbf{T}$ =构成的前馈神经网络(FFN)，将解码器嵌入**f**中包含的位移信息映射到同一空间中的线性投影：FFN(**f**)。

条件空间查询是通过在嵌入空间中对参考点进行变换来计算的：$\mathbf{p}_q=\mathbf{T}\mathbf{p}_s$。我们选择了简单且计算高效的投影矩阵——一个对角矩阵。其256个对角元素表示为一个向量$\boldsymbol{\lambda}_q$。条件空间查询通过逐元素相乘计算得出：

$$\mathbf{p}_q=\mathbf{T}\mathbf{p}_s=\boldsymbol{\lambda}_q\odot\mathbf{p}_s.\tag{6}$$

多头交叉注意力。遵循DETR[3]的做法，我们采用了标准的多头交叉注意力机制。物体检测通常需要隐式或显式地定位物体的四个边界极值点以实现精确的边界框回归，并定位物体区域以实现准确的物体分类。多头机制有助于解耦这些定位任务。

我们通过将查询、键和值M =用学习到的线性投影8次映射到低维空间，实现多头并行注意力机制。空间查询（键）与内容查询（键）分别通过不同的线性投影独立映射至每个注意力头。值的投影方式与原始DETR保持一致，仅针对内容部分进行映射。

3.4. 可视化与分析

可视化。图4展示了每个注意力头的权重图：空间注意力权重图、内容注意力权重图以及组合注意力权重图。这些权重图分别基于空间点积$\mathbf{p}_q^\top\mathbf{p}_k$、内容点积$\mathbf{c}_q^\top\mathbf{c}_k$和组合点积$\mathbf{c}_q^\top\mathbf{c}_k+\mathbf{p}_q^\top\mathbf{p}_k$进行了soft-max归一化处理。我们展示了8个图中的5个，其余三个为重复项，分别对应物体框[4]的底部与顶部边缘区域以及内部一小块区域。

[4]The duplicates might be different for models trained several times, but the detection performance is almost the same.

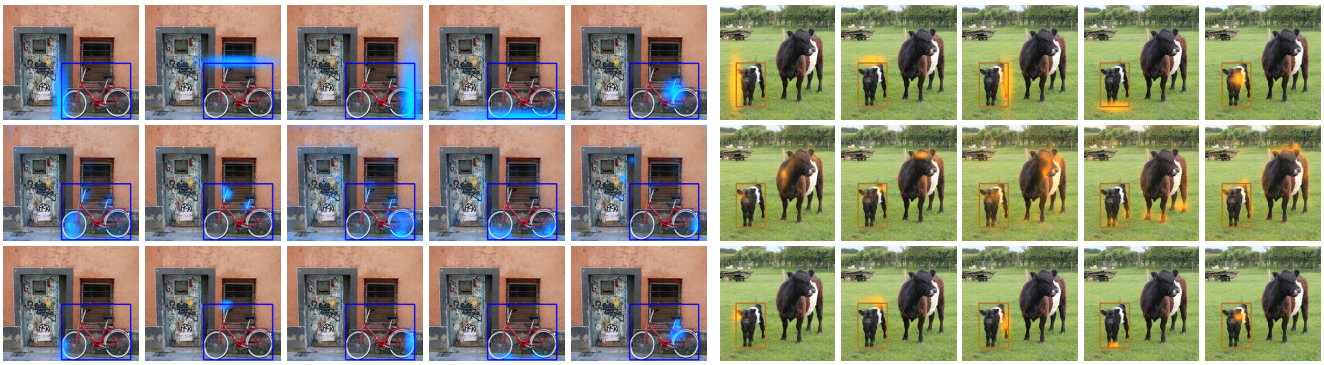

Figure 4. Illustrating the spatial attention weight maps (the first row), the content attention weight maps (the second row), and the combined attention weight maps (the third row) computed from our conditional DETR. The attention weight maps are from 5 heads out of the 8 heads and are responsible for the four extremities and a region inside the object box. The content attention weight maps for the four extremities highlight scattered regions inside the box (bicycle) or similar regions in two object instances (cow), and the corresponding combined attention weight maps highlight the extremity regions with the help of the spatial attention weight maps. The combined attention weight map for the region inside the object box mainly depends on the spatial attention weight map, which implies that the representation of a region inside the object might encode enough class information. The maps are from conditional DETR-R50 trained with 50 epochs.

We can see that the spatial attention weight map at each head is able to localize a distinct region, a region containing one extremity or a region inside the object box. It is interesting that each spatial attention weight map corresponding to an extremity highlights a spatial band that overlaps with the corresponding edge of the object box. The other spatial attention map for the region inside the object box merely highlights a small region whose representations might already encode enough information for object classification.

The content attention weight maps of the four heads corresponding to the four extremities highlight scattered regions in addition to the extremities. The combination of the spatial and content maps filters out other highlights and keeps extremity highlights for accurate box regression.

Comparison to DETR. Figure 1 shows the spatial attention weight maps of our conditional DETR (the first row) and the original DETR trained with 50 epochs (the second row). The maps of our approach are computed by soft-max normalizing the dot-products between spatial keys and queries, $\mathbf{p}_q^\top \mathbf{p_k}$. The maps for DETR are computed by soft-max normalizing the dot-products with the spatial keys, $(\mathbf{o}_q + \mathbf{c}_q)^\top \mathbf{p}_k$.

It can be seen that our spatial attention weight maps accurately localize the distinct regions, four extremities. In contrast, the maps from the original DETR with 50 epochs can not accurately localize two extremities, and 500 training epochs (the third row) make the content queries stronger, leading to accurate localization. This implies that it is really hard to learn the content query $\mathbf{c}_q$ to serve as two roles[5]: match the content key and the spatial key simultaneously, and thus more training epochs are needed.

[5]Strictly speaking, the embedding output from decoder self-attention for more training epochs contains both spatial and content information. For discussion convenience, we still call it content query.

Analysis. The spatial attention weight maps shown in Figure 4 imply that the conditional spatial query, used to form the spatial query, have at least two effects. (i) Translate the highlight positions to the four extremities and the position inside the object box: interestingly the highlighted positions are spatially similarly distributed in the object box. (ii) Scale the spatial spread for the extremity highlights: large spread for large objects and small spread for small objects.

The two effects are realized in the spatial embedding space through applying the transformation $\mathbf{T}$ over $\mathbf{p}_s$ (further disentangled through image-independent linear projections contained in cross-attention and distributed to each head). This indicates that the transformation $\mathbf{T}$ not only contains the displacements as discussed before, but also the object scale.

3.5. Implementation Details

Architecture. Our architecture is almost the same with the DETR architecture [3] and contains the CNN backbone, transformer encoder, transformer decoder, prediction feed-forward networks (FFNs) following each decoder layer (the last decoder layer and the 5 internal decoder layers) with parameters shared among the 6 prediction FFNs. The hyper-parameters are the same as DETR.

The main architecture difference is that we introduce the conditional spatial embeddings as the spatial queries for conditional multi-head cross-attention and that the spatial query (key) and the content query (key) are combined through concatenation other than addition. In the first cross-attention layer there are no decoder content embeddings, we make simple changes based on the DETR implementation [3]: concatenate the positional embedding predicted from the object query (the positional embedding) into the original query (key).

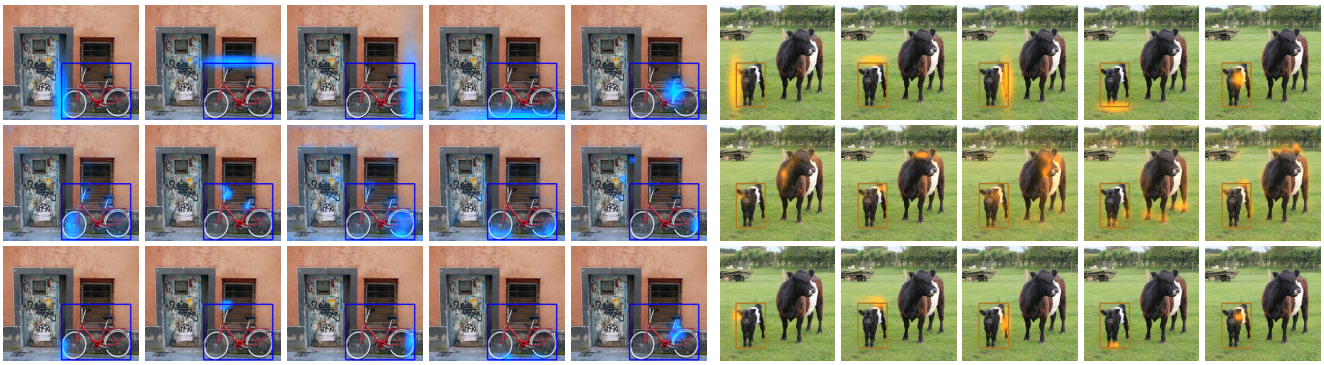

图4展示了由我们的条件DETR计算得到的空间注意力权重图（第一行）、内容注意力权重图（第二行）及组合注意力权重图（第三行）。这些注意力权重图选自8个头中的5个，分别负责物体框的四个边缘区域及一个内部区域。针对四个边缘区域的内容注意力权重图会突出框内分散区域（自行车）或两个物体实例中的相似区域（牛），而相应的组合注意力权重图则在空间注意力权重图的辅助下精准锁定边缘区域。对于物体框内部区域的组合注意力权重图则主要依赖于空间注意力权重图，这表明物体内部区域的表征可能已编码足够的类别信息。所有图示均来自训练50个周期的条件DETR-R50模型。

我们可以观察到，每个注意力头生成的空间注意力权重图都能定位到一个独特区域，这个区域要么包含一个肢体末端，要么位于物体框内部。有趣的是，对应肢体末端的空间注意力权重图会突出显示与物体框相应边缘重叠的空间带状区域。而针对物体框内部区域的另一张空间注意力图，则仅会突显一个小范围区域——该区域的表征可能已编码足够信息以支撑物体分类任务。

四个头部对应的内容注意力权重图除了突出四肢区域外，还强调了分散的区域。空间图与内容图的结合过滤掉了其他高亮部分，保留了四肢的高亮显示，以实现精确的边界框回归。

与DETR的对比。图1展示了我们的条件式DETR（第一行）与原版DETR经过50轮训练（第二行）的空间注意力权重图。我们方法中的权重图通过空间键与查询的点积经soft-max归一化计算得出，$\mathbf{p}_q^\top \mathbf{p_k}$。而DETR的权重图则是对空间键的点积进行soft-max归一化处理，$(\mathbf{o}_q + \mathbf{c}_q)^\top \mathbf{p}_k$。

可以看出，我们的空间注意力权重图精准定位了四肢等不同区域。相比之下，原始DETR模型经过50轮训练生成的定位图无法准确标定两处肢体末端，而500轮训练（第三行）则使内容查询$\mathbf{c}_q$更具判别力，从而实现精确定位。这表明要让内容查询$\mathbf{c}_q$同时承担匹配内容键与空间键[5]的双重角色十分困难，因此需要更长的训练周期。

[5]Strictly speaking, the embedding output from decoder self-attention for more training epochs contains both spatial and content information. For discussion convenience, we still call it content query.

分析。图4所示的空间注意力权重图表明，用于构建空间查询的条件性空间查询至少产生两种效果。(i) 将高亮位置平移至物体框的四个边缘及内部区域：值得注意的是，高亮位置在物体框内的空间分布呈现相似性。(ii) 根据物体尺寸调节边缘高亮区域的空间扩展范围：大物体对应大范围扩展，小物体则对应小范围扩展。

这两种效应通过在空间嵌入空间中对$\mathbf{p}_s$ (应用变换$\mathbf{T}$得以实现，并通过包含在交叉注意力中、分发至每个头)的图像无关线性投影进一步解耦。这表明变换$\mathbf{T}$不仅包含先前讨论的位移，还包含了物体尺度。

3.5. 实现细节

架构。我们的架构与DETR架构[3]几乎相同，包含CNN主干网络、Transformer编码器、Transformer解码器，以及跟随每个解码器层（最后一层解码器及内部5层解码器）的预测前馈网络（FFNs），这6个预测FFNs共享参数。超参数设置与DETR保持一致。

主要架构差异在于，我们引入了条件空间嵌入作为条件多头交叉注意力的空间查询，并且空间查询（键）与内容查询（键）通过拼接而非相加的方式结合。在首个交叉注意力层中不存在解码器内容嵌入，我们基于DETR实现[3]进行了简单调整：将对象查询预测的位置嵌入（即位置编码）拼接到原始查询（键）中。

Table 1. Comparison of conditional DETR with DETR on COCO 2017 `val`. Our conditional DETR approach for high-resolution backbones DC5-R50 and DC5-R101 is $10\times$ faster than the original DETR, and for low-resolution backbones R50 and R101 $6.67\times$ faster. Conditional DETR is empirically superior to other two single-scale DETR variants. *The results of deformable DETR are from the GitHub repository provided by the authors of deformable DETR [53].

Model	#epochs	GFLOPs	#params (M)	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
DETR-R50	500	86	41	42.0	62.4	44.2	20.5	45.8	61.1
DETR-R50	50	86	41	34.9	55.5	36.0	14.4	37.2	54.5
Conditional DETR-R50	50	90	44	40.9	61.8	43.3	20.8	44.6	59.2
Conditional DETR-R50	75	90	44	42.1	62.9	44.8	21.6	45.4	60.2
Conditional DETR-R50	108	90	44	43.0	64.0	45.7	22.7	46.7	61.5
DETR-DC5-R50	500	187	41	43.3	63.1	45.9	22.5	47.3	61.1
DETR-DC5-R50	50	187	41	36.7	57.6	38.2	15.4	39.8	56.3
Conditional DETR-DC5-R50	50	195	44	43.8	64.4	46.7	24.0	47.6	60.7
Conditional DETR-DC5-R50	75	195	44	44.5	65.2	47.3	24.4	48.1	62.1
Conditional DETR-DC5-R50	108	195	44	45.1	65.4	48.5	25.3	49.0	62.2
DETR-R101	500	152	60	43.5	63.8	46.4	21.9	48.0	61.8
DETR-R101	50	152	60	36.9	57.8	38.6	15.5	40.6	55.6
Conditional DETR-R101	50	156	63	42.8	63.7	46.0	21.7	46.6	60.9
Conditional DETR-R101	75	156	63	43.7	64.9	46.8	23.3	48.0	61.7
Conditional DETR-R101	108	156	63	44.5	65.6	47.5	23.6	48.4	63.6
DETR-DC5-R101	500	253	60	44.9	64.7	47.7	23.7	49.5	62.3
DETR-DC5-R101	50	253	60	38.6	59.7	40.7	17.2	42.2	57.4
Conditional DETR-DC5-R101	50	262	63	45.0	65.5	48.4	26.1	48.9	62.8
Conditional DETR-DC5-R101	75	262	63	45.6	66.5	48.8	25.5	49.7	63.3
Conditional DETR-DC5-R101	108	262	63	45.9	66.8	49.5	27.2	50.3	63.3
Other single-scale DETR variants									
Deformable DETR-R50-SS*	50	78	34	39.4	59.6	42.3	20.6	43.0	55.5
UP-DETR-R50 [5]	150	86	41	40.5	60.8	42.6	19.0	44.4	60.0
UP-DETR-R50 [5]	300	86	41	42.8	63.0	45.3	20.8	47.1	61.7
Deformable DETR-DC5-R50-SS*	50	128	34	41.5	61.8	44.9	24.1	45.3	56.0

Reference points. In the original DETR approach, $\mathbf{s} = [0\ 0]^\top$ is the same for all the decoder embeddings. We study two ways forming the reference points: regard the unnormalized 2D coordinates as learnable parameters, and the unnormalized 2D coordinate predicted from the object query $\mathbf{o}_q$. In the latter way that is similar to deformable DETR [53], the prediction unit is an FFN and consists of learnable linear projection + ReLU + learnable linear projection: $\mathbf{s} = \mathrm{FFN}(\mathbf{o}_q)$. When used for forming the conditional spatial query, the 2D coordinates are normalized by the sigmoid function.

Loss function. We follow DETR [3] to find an optimal bipartite matching [20] between the predicted and ground-truth objects using the Hungarian algorithm, and then form the loss function for computing and back-propagate the gradients. We use the same way with deformable DETR [53] to formulate the loss: the same matching cost function, the same loss function with 300 object queries, and the same trade-off parameters; The classification loss function is focal loss [24], and the box regression loss (including L1 and GIoU [34] loss) is the same as DETR [3].

4. Experiments

4.1. Setting

Dataset. We perform the experiments on the COCO 2017 [25] detection dataset. The dataset contains about 118K training images and 5K validation (`val`) images.

Training. We follow the DETR training protocol [3]. The backbone is the ImageNet-pretrained model from TORCHVISION with batchnorm layers fixed, and the transformer parameters are initialized using the Xavier initialization scheme [10]. The weight decay is set to be 10^{-4}. The AdamW [27] optimizer is used. The learning rates for the backbone and the transformer are initially set to be 10^{-5} and 10^{-4}, respectively. The dropout rate in transformer is 0.1. The learning rate is dropped by a factor of 10 after 40 epochs for 50 training epochs, after 60 epochs for 75 training epochs, and after 80 epochs for 108 training epochs.

We use the augmentation scheme same as DETR [3]: resize the input image such that the short side is at least 480 and at most 800 pixels and the long size is at most 1333 pixels; randomly crop the image such that a training image is

表1. 条件DETR与DETR在COCO 2017验证集上的对比。我们针对高分辨率骨干网络DC5-R50和DC5-R101提出的条件DETR方法比原始DETR快10×，针对低分辨率骨干网络R50和R101则快6.67×。实验表明，条件DETR优于其他两种单尺度DETR变体。* 可变形DETR的结果来自其作者在GitHub仓库提供的数据[53]。

Model	#epochs	GFLOPs	#params (M)	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
DETR-R50	500	86	41	42.0	62.4	44.2	20.5	45.8	61.1
DETR-R50	50	86	41	34.9	55.5	36.0	14.4	37.2	54.5
Conditional DETR-R50	50	90	44	40.9	61.8	43.3	20.8	44.6	59.2
Conditional DETR-R50	75	90	44	42.1	62.9	44.8	21.6	45.4	60.2
Conditional DETR-R50	108	90	44	43.0	64.0	45.7	22.7	46.7	61.5
DETR-DC5-R50	500	187	41	43.3	63.1	45.9	22.5	47.3	61.1
DETR-DC5-R50	50	187	41	36.7	57.6	38.2	15.4	39.8	56.3
Conditional DETR-DC5-R50	50	195	44	43.8	64.4	46.7	24.0	47.6	60.7
Conditional DETR-DC5-R50	75	195	44	44.5	65.2	47.3	24.4	48.1	62.1
Conditional DETR-DC5-R50	108	195	44	45.1	65.4	48.5	25.3	49.0	62.2
DETR-R101	500	152	60	43.5	63.8	46.4	21.9	48.0	61.8
DETR-R101	50	152	60	36.9	57.8	38.6	15.5	40.6	55.6
Conditional DETR-R101	50	156	63	42.8	63.7	46.0	21.7	46.6	60.9
Conditional DETR-R101	75	156	63	43.7	64.9	46.8	23.3	48.0	61.7
Conditional DETR-R101	108	156	63	44.5	65.6	47.5	23.6	48.4	63.6
DETR-DC5-R101	500	253	60	44.9	64.7	47.7	23.7	49.5	62.3
DETR-DC5-R101	50	253	60	38.6	59.7	40.7	17.2	42.2	57.4
Conditional DETR-DC5-R101	50	262	63	45.0	65.5	48.4	26.1	48.9	62.8
Conditional DETR-DC5-R101	75	262	63	45.6	66.5	48.8	25.5	49.7	63.3
Conditional DETR-DC5-R101	108	262	63	45.9	66.8	49.5	27.2	50.3	63.3
Other single-scale DETR variants									
Deformable DETR-R50-SS*	50	78	34	39.4	59.6	42.3	20.6	43.0	55.5
UP-DETR-R50 [5]	150	86	41	40.5	60.8	42.6	19.0	44.4	60.0
UP-DETR-R50 [5]	300	86	41	42.8	63.0	45.3	20.8	47.1	61.7
Deformable DETR-DC5-R50-SS*	50	128	34	41.5	61.8	44.9	24.1	45.3	56.0

参考点。在原始DETR方法中，$\mathbf{s} = [0\ 0]^\top$对所有解码器嵌入是相同的。我们研究了两种形成参考点的方式：将未归一化的2D坐标视为可学习参数，以及由对象查询$\mathbf{o}_q$预测出的未归一化2D坐标。后一种方式类似于可变形DETR[53]，其预测单元是一个FFN，由可学习的线性投影+、ReLU激活+及可学习的线性投影组成：$\mathbf{s} = \text{FFN}(\mathbf{o}_q)$。当用于构建条件空间查询时，2D坐标通过sigmoid函数进行归一化处理。

损失函数。我们遵循DETR[3]的方法，利用匈牙利算法在预测目标与真实目标之间寻找最优二分匹配[20]，进而构建损失函数以计算并反向传播梯度。我们采用与可变形DETR[53]相同的损失构建方式：使用相同的匹配代价函数、包含300个对象查询的相同损失函数及相同的权衡参数；分类损失函数采用焦点损失[24]，而边界框回归损失（包含L1损失和GIoU[34]损失）则与DETR[3]保持一致。

4. 实验

4.1. 设置

数据集。我们在COCO 2017 [25]检测数据集上进行实验。该数据集包含约118K训练图像和5K验证（val）图像。

训练。我们遵循DETR的训练方案[3]。骨干网络采用TORCHVISION中经过ImageNet预训练的模型，并固定其批归一化层，而Transformer参数则采用Xavier初始化方案[10]进行初始化。权重衰减设置为10^{-4}。优化器选用AdamW[27]。骨干网络和Transformer的初始学习率分别设为10^{-5}和10^{-4}。Transformer中的丢弃率为0.1。学习率在训练过程中会分阶段下降：50个训练周期时在第40周期后降至1/10，75个周期时在第60周期后下降，108个周期时则在第80周期后进行同样的衰减。

我们采用与DETR[3]相同的增强方案：调整输入图像尺寸，使短边至少为480像素且不超过800像素，长边不超过1333像素；随机裁剪图像以确保训练图像

Table 2. Results for multi-scale and higher-resolution DETR variants. We do not expect that our approach performs on par as our approach (single-scale, 16× resolution) does not use a strong multi-scale or 8× resolution encoder. Surprisingly, the AP scores of our approach with DC5-R50 and DC5-R101 are close to the two multi-scale and higher-resolution DETR variants.

Model	#epochs	GFLOPs	#params (M)	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster RCNN-FPN-R50 [33]	36	180	42	40.2	61.0	43.8	24.2	43.5	52.0
Faster RCNN-FPN-R50 [33]	108	180	42	42.0	62.1	45.5	26.6	45.5	53.4
Deformable DETR-R50 [53]	50	173	40	43.8	62.6	47.7	26.4	47.1	58.0
TSP-FCOS-R50 [37]	36	189	–	43.1	62.3	47.0	26.6	46.8	55.9
TSP-RCNN-R50 [37]	36	188	–	43.8	63.3	48.3	28.6	46.9	55.7
TSP-RCNN-R50 [37]	96	188	–	45.0	64.5	49.6	29.7	47.7	58.0
Conditional DETR-DC5-R50	50	195	44	43.8	64.4	46.7	24.0	47.6	60.7
Conditional DETR-DC5-R50	108	195	44	45.1	65.4	48.5	25.3	49.0	62.2
Faster RCNN-FPN-R101 [33]	36	246	60	42.0	62.5	45.9	25.2	45.6	54.6
Faster RCNN-FPN-R101 [33]	108	246	60	44.0	63.9	47.8	27.2	48.1	56.0
TSP-FCOS-R101 [37]	36	255	–	44.4	63.8	48.2	27.7	48.6	57.3
TSP-RCNN-R101 [37]	36	254	–	44.8	63.8	49.2	29.0	47.9	57.1
TSP-RCNN-R101 [37]	96	254	–	46.5	66.0	51.2	29.9	49.7	59.2
Conditional DETR-DC5-R101	50	262	63	45.0	65.5	48.4	26.1	48.9	62.8
Conditional DETR-DC5-R101	108	262	63	45.9	66.8	49.5	27.2	50.3	63.3

cropped with probability 0.5 to a random rectangular patch.

Evaluation. We use the standard COCO evaluation. We report the average precision (AP), and the AP scores at 0.50, 0.75 and for the small, medium, and large objects.

4.2. Results

Comparison to DETR. We compare the proposed conditional DETR to the original DETR [3]. We follow [3] and report the results over four backbones: ResNet-50 [12], ResNet-101, and their 16×-resolution extensions DC5-ResNet-50 and DC5-ResNet-101.

The corresponding DETR models are named as DETR-R50, DETR-R101, DETR-DC5-R50, and DETR-DC5-R101, respectively. Our models are named as conditional DETR-R50, conditional DETR-R101, conditional DETR-DC5-R50, and conditional DETR-DC5-R101, respectively.

Table 1 presents the results from DETR and conditional DETR. DETR with 50 training epochs performs much worse than 500 training epochs. Conditional DETR with 50 training epochs for R50 and R101 as the backbones performs slightly worse than DETR with 500 training epochs. Conditional DETR with 50 training epochs for DC5-R50 and DC5-R101 performs similarly as DETR with 500 training epochs. Conditional DETR for the four backbones with 75/108 training epochs performs better than DETR with 500 training epochs. In summary, conditional DETR for high-resolution backbones DC5-R50 and DC5-R101 is 10× faster than the original DETR, and for low-resolution backbones R50 and R101 6.67× faster. In other words, conditional DETR performs better for stronger backbones with better performance.

In addition, we report the results of single-scale DETR extensions: deformable DETR-SS [53] and UP-DETR [5] in Table 1. Our results over R50 and DC5-R50 are better than deformable DETR-SS: 40.9 vs. 39.4 and 43.8 vs. 41.5. The comparison might not be fully fair as for example parameter and computation complexities are different, but it implies that the conditional cross-attention mechanism is beneficial. Compared to UP-DETR-R50, our results with fewer training epochs are obviously better.

Comparison to multi-scale and higher-resolution DETR variants. We focus on accelerating the DETR training, without addressing the issue of high computational complexity in the encoder. We do not expect that our approach achieves on par with DETR variants w/ multi-scale attention and 8×-resolution encoders, e.g., TSP-FCOS and TSP-RCNN [37] and deformable DETR [53], which are able to reduce the encoder computational complexity and improve the performance due to multi-scale and higher-resolution.

The comparisons in Table 2 surprisingly show that our approach on DC5-R50 (16×) performs same as deformable DETR-R50 (multi-scale, 8×). Considering that the AP of the single-scale deformable DETR-DC5-R50-SS is 41.5 (lower than ours 43.8) (Table 1), one can see that deformable DETR benefits a lot from the multi-scale and higher-resolution encoder that potentially benefit our approach, which is currently not our focus and left as our future work.

The performance of our approach is also on par with TSP-FCOS and TSP-RCNN. The two methods contain a transformer encoder over a small number of selected positions/regions (feature of interest in TSP-FCOS and region proposals in TSP-RCNN) without using the transformer decoder, are extensions of FCOS [39] and Faster RCNN [33].

表2. 多尺度及高分辨率DETR变体的结果。我们并未预期本方法（单尺度，16×分辨率）能与采用强力多尺度或8×分辨率编码器的方案表现相当。但出乎意料的是，采用DC5-R50和DC5-R101的本方法AP分数与两种多尺度高分辨率DETR变体十分接近。

Model	#epochs	GFLOPs	#params (M)	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster RCNN-FPN-R50 [33]	36	180	42	40.2	61.0	43.8	24.2	43.5	52.0
Faster RCNN-FPN-R50 [33]	108	180	42	42.0	62.1	45.5	26.6	45.5	53.4
Deformable DETR-R50 [53]	50	173	40	43.8	62.6	47.7	26.4	47.1	58.0
TSP-FCOS-R50 [37]	36	189	–	43.1	62.3	47.0	26.6	46.8	55.9
TSP-RCNN-R50 [37]	36	188	–	43.8	63.3	48.3	28.6	46.9	55.7
TSP-RCNN-R50 [37]	96	188	–	45.0	64.5	49.6	29.7	47.7	58.0
Conditional DETR-DC5-R50	50	195	44	43.8	64.4	46.7	24.0	47.6	60.7
Conditional DETR-DC5-R50	108	195	44	45.1	65.4	48.5	25.3	49.0	62.2
Faster RCNN-FPN-R101 [33]	36	246	60	42.0	62.5	45.9	25.2	45.6	54.6
Faster RCNN-FPN-R101 [33]	108	246	60	44.0	63.9	47.8	27.2	48.1	56.0
TSP-FCOS-R101 [37]	36	255	–	44.4	63.8	48.2	27.7	48.6	57.3
TSP-RCNN-R101 [37]	36	254	–	44.8	63.8	49.2	29.0	47.9	57.1
TSP-RCNN-R101 [37]	96	254	–	46.5	66.0	51.2	29.9	49.7	59.2
Conditional DETR-DC5-R101	50	262	63	45.0	65.5	48.4	26.1	48.9	62.8
Conditional DETR-DC5-R101	108	262	63	45.9	66.8	49.5	27.2	50.3	63.3

以0.5的概率随机裁剪为一个矩形区域。

评估。我们采用标准的COCO评估方法，报告平均精度（AP）以及在不同IoU阈值下的AP分数，包括0.50、0.75，并针对小、中、大尺寸目标分别给出结果。

4.2. 结果

与DETR的对比。我们将提出的条件DETR与原始DETR[3]进行比较。遵循[3]的方法，我们报告了四种主干网络的结果：ResNet-50[12]、ResNet-101，以及它们16×分辨率的扩展版本DC5-ResNet-50和DC5-ResNet-101。

对应的DETR模型分别命名为DETR-R50、DETR-R101、DETR-DC5-R50和DETR-DC5-R101。我们的模型则分别命名为conditional DETR-R50、conditional DETR-R101、conditional DETR-DC5-R50和conditional DETR-DC5-R101。

表1展示了DETR与条件DETR的实验结果。采用50训练轮次的DETR性能远逊于500轮次版本。以R50和R101为骨干网络的条件DETR在50轮次训练下，表现略低于500轮次的DETR；而采用DC5-R50和DC5-R101骨干时，50轮次条件DETR与500轮次DETR性能相当。当四种骨干网络在75/108训练轮次下，条件DETR全面超越500轮次DETR。总体而言，针对高分辨率骨干DC5-R50和DC5-R101，条件DETR比原始DETR提速10×倍；对低分辨率骨干R50和R101则实现6.67×倍加速。换言之，条件DETR在性能更强的骨干网络上表现更优。

此外，我们报告了单尺度DETR的结果扩展：表1中的可变形DETR-SS [53]和UP-DETR [5]。我们在R50和DC5-R50上的结果优于可变形DETR-SS：40.9对比39.4，以及43.8对比41.5。这一比较可能不完全公平，例如参数和计算复杂度存在差异，但它暗示了条件交叉注意力机制的有效性。与UP-DETR-R50相比，我们以更少的训练周期取得了明显更优的结果。

与多尺度及高分辨率DETR变体的比较。我们专注于加速DETR训练，而未解决编码器中高计算复杂度的问题。我们不期望我们的方法能达到与采用多尺度注意力及8×分辨率编码器的DETR变体（如TSP-FCOS、TSP-RCNN[37]和可变形DETR[53]）相当的水平，这些变体能够通过多尺度和更高分辨率降低编码器计算复杂度并提升性能。

表2中的比较结果令人惊讶地显示，我们的DC5-R50方法（16×）与可变形DETR-R50（多尺度，8×）表现相当。考虑到单尺度可变形DETR-DC5-R50-SS的AP为41.5（低于我们的43.8）（表1），可以看出可变形DETR从多尺度和更高分辨率的编码器中获益良多，这些特性也可能使我们的方法受益，但目前并非我们的研究重点，留作未来工作。

我们方法的性能也与TSP-FCOS和TSP-RCNN相当。这两种方法在少量选定的位置/区域（TSP-FCOS中的兴趣特征和TSP-RCNN中的区域提议）上使用了transformer编码器，而未采用transformer解码器，是对FCOS[39]和Faster RCNN[33]的扩展。

Table 3. Ablation study for the ways forming the conditional spatial query. CSQ = our proposed conditional spatial query scheme. Please see the first two paragraphs in Section 5.3 for the meanings of CSQ variants. Our proposed CSQ manner performs better. The backbone ResNet-50 is adopted.

Exp.	CSQ-C	CSQ-T	CSQ-P	CSQ-I	CSQ
GFLOPs	89.3	89.5	89.3	89.5	89.5
AP	37.1	37.6	37.8	40.2	40.9

It should be noted that position/region selection removes unnecessary computation in self-attention and reduces computation complexity dramatically.

4.3. Ablations

Reference points. We compare three ways of forming reference points $\mathbf{s}$: (i) $\mathbf{s} = (0, 0)$, same to the original DETR, (ii) learn $\mathbf{s}$ as model parameters and each prediction is associated with different reference points, and (iii) predict each reference point $\mathbf{s}$ from the corresponding object query. We conducted the experiments with ResNet-50 as the backbone. The AP scores are 36.8, 40.7, and 40.9, suggesting that (ii) and (iii) perform on par and better than (i).

The effect of the way forming the conditional spatial query. We empirically study how the transformation $\boldsymbol{\lambda}_q$ and the positional embedding $\mathbf{p}_s$ of the reference point, used to form the conditional spatial query $\mathbf{p}_q = \boldsymbol{\lambda}_q \odot \mathbf{p}_s$, make contributions to the detection performance.

We report the results of our conditional DETR, and the other ways forming the spatial query with: (i) CSQ-P - only the positional embedding $\mathbf{p}_s$, (ii) CSQ-T - only the transformation $\boldsymbol{\lambda}_q$, (iii) CSQ-C - the decoder content embedding $\mathbf{f}$, and (iv) CSQ-I - the element-wise product of the transformation predicted from the decoder self-attention output $\mathbf{c}_q$ and the positional embedding $\mathbf{p}_s$. The studies in Table 3 imply that our proposed way (CSQ) performs overall the best, validating our analysis about the transformation predicted from the decoder embedding and the positional embedding of the reference point in Section 3.3.

Focal loss and offset regression with respect to learned reference point. Our approach follows deformable DETR [53]: use the focal loss with 300 object queries to form the classification loss and predict the box center by regressing the offset with respect to the reference point. We report how the two schemes affect the DETR performance in Table 4. One can see that separately using the focal loss or center offset regression without learning refererence points leads to a slight AP gain and combining them together leads to a larger AP gain. Conditional cross-attention in our approach built on the basis of focal loss and offset regression brings a major gain 4.0.

The effect of linear projections $\mathbf{T}$ forming the transformation. Predicting the conditional spatial query needs to learn the linear projection $\mathbf{T}$ from the decoder embedding

Table 4. The empirical results about the focal loss (FL), offset regression (OR) for box center prediction, and our conditional spatial query (CSQ). The backbone ResNet-50 is adopted.

OR	FL	CSQ	GFLOPs	AP
			85.5	34.9
✓			85.5	35.0
	✓		88.1	35.3
✓	✓		88.1	36.9
✓	✓	✓	89.5	40.9

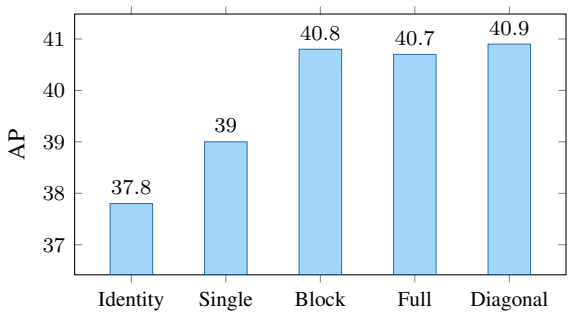

Figure 5. The empirical results for different forms of linear projections that are used to compute the spatial queries for conditional multi-head cross-attention. Diagonal (ours), Full, and Block perform on par. The backbone ResNet-50 is adopted.

(see Equation 6). We empirically study how the linear projection forms affect the performance. The linear projection forms include: an *identity* matrix that means not to learn the linear projection, a *single* scalar, a *block* diagonal matrix meaning that each head has a learned 32×32 linear projection matrix, a *full* matrix without constraints, and a *diagonal* matrix. Figure 5 presents the results. It is interesting that a single-scalar helps improve the performance, maybe due to narrowing down the spatial range to the object area. Other three forms, *block* diagonal, *full*, and *diagonal* (ours), perform on par.

5. Conclusion

We present a simple conditional cross-attention mechanism. The key is to learn a spatial query from the corresponding reference point and decoder embedding. The spatial query contains the spatial information mined for the class and box prediction in the previous decoder layer, and leads to spatial attention weight maps highlighting the bands containing extremities and small regions inside the object box. This shrinks the spatial range for the content query to localize the distinct regions, thus relaxing the dependence on the content query and reducing the training difficulty. In the future, we will study the proposed conditional cross-attention mechanism for human pose estimation [8, 41, 36] and line segment detection [43].

Acknowledgments. We thank the anonymous reviewers for their insightful comments and suggestions on our manuscript.

表3. 条件空间查询形成方式的消融研究。CSQ {v*} 为我们提出的条件空间查询方案。关于CSQ变体的含义，请参阅第5.3节的前两段。我们提出的CSQ方式表现更优。采用ResNet-50作为骨干网络。

Exp.	CSQ-C	CSQ-T	CSQ-P	CSQ-I	CSQ
GFLOPs	89.3	89.5	89.3	89.5	89.5
AP	37.1	37.6	37.8	40.2	40.9

需要注意的是，位置/区域选择能够消除自注意力机制中不必要的计算，从而显著降低计算复杂度。

4.3. 消融实验

参考点。我们比较了三种构建参考点$\mathbf{s}$的方法：(i) $\mathbf{s}=(0,0)$，与原版DETR相同；(ii) 将$\mathbf{s}$作为模型参数学习，每个预测关联不同的参考点；(iii) 从对应的对象查询中预测每个参考点$\mathbf{s}$。实验采用ResNet-50作为骨干网络，AP得分分别为36.8、40.7和40.9，表明方法(ii)和(iii)表现相当且优于方法(i)。

形成条件空间查询方式的影响。我们实证研究了用于构建条件空间查询$\mathbf{p}_q = \boldsymbol{\lambda}_q \odot \mathbf{p}_s$的参考点变换$\boldsymbol{\lambda}_q$和位置嵌入$\mathbf{p}_s$如何对检测性能做出贡献。

我们报告了条件DETR的结果，以及其他几种构建空间查询的方式：(i) CSQ-P - 仅使用位置嵌入$\mathbf{p}_s$，(ii) CSQ-T - 仅使用变换$\boldsymbol{\lambda}_q$，(iii) CSQ-C - 解码器内容嵌入$\mathbf{f}$，以及(iv) CSQ-I - 解码器自注意力输出预测的变换$\mathbf{c}_q$与位置嵌入$\mathbf{p}_s$的逐元素乘积。表3中的研究表明，我们提出的方法(CSQ)总体表现最佳，验证了我们在3.3节中关于从解码器嵌入和参考点位置嵌入预测变换的分析。

聚焦损失与基于学习参考点的偏移回归。我们的方法遵循可变形DETR[53]：采用300个物体查询的聚焦损失构建分类损失，并通过相对于参考点的偏移回归预测框中心。表4展示了这两种方案对DETR性能的影响。可见，单独使用聚焦损失或中心偏移回归而不学习参考点时AP略有提升，而二者结合则带来更大的AP增益。在我们的方法中，建立在聚焦损失和偏移回归基础上的条件交叉注意力机制带来了显著的4.0提升。

线性投影$\mathbf{T}$构成了变换的效果。预测条件空间查询需要从解码器嵌入中学习线性投影$\mathbf{T}$

表4. 关于焦点损失(FL)、用于框中心预测的偏移回归(OR)以及我们的条件空间查询(CSQ)的实证结果。采用ResNet-50作为主干网络。

OR	FL	CSQ	GFLOPs	AP
			85.5	34.9
✓			85.5	35.0
	✓		88.1	35.3
✓	✓		88.1	36.9
✓	✓	✓	89.5	40.9

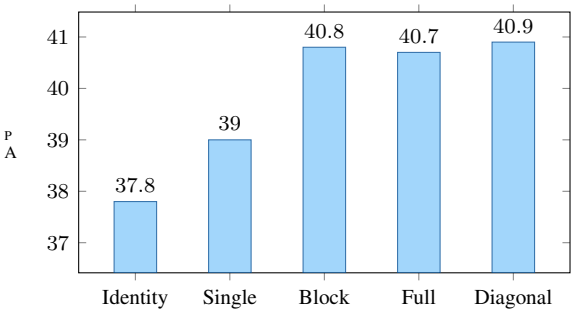

图5. 用于计算条件多头交叉注意力空间查询的不同线性投影形式的实证结果。对角线（我们的方法）、全矩阵和分块表现相当。采用ResNet-50作为骨干网络。

（见公式6）。我们实证研究了线性投影形式如何影响性能。线性投影形式包括：一个*identity*矩阵（表示不学习线性投影）、一个*single*标量、一个*block*对角矩阵（意味着每个头都有一个学习到的32×32线性投影矩阵）、一个*full*无约束矩阵，以及一个*diagonal*矩阵。图5展示了结果。有趣的是，单一标量有助于提升性能，这可能是由于将空间范围缩小至目标区域所致。其他三种形式——*block*对角、*full*以及*diagonal*(ours)——表现相当。

5. 结论

我们提出了一种简单的条件交叉注意力机制。其核心在于从对应的参考点与解码器嵌入中学习空间查询。该空间查询蕴含了先前解码器层为类别和边界框预测所挖掘的空间信息，并生成空间注意力权重图，突出显示包含物体框内肢体末端及小区域的频带。这缩小了内容查询的空间范围，以定位独特区域，从而降低了对内容查询的依赖，减轻了训练难度。未来，我们将研究该条件交叉注意力机制在人体姿态估计[8, 41, 36]和线段检测[43]中的应用。

致谢。我们感谢匿名审稿人对我们稿件提出的深刻见解和建议。

References

[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *CoRR*, abs/2004.10934, 2020. 2

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: delving into high quality object detection. In *CVPR*, 2018. 2

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1, 2, 3, 4, 5, 6, 7

[4] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *CVPR*, 2020. 2

[5] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. UP-DETR: unsupervised pre-training for object detection with transformers. *CoRR*, abs/2011.09094, 2020. 6, 7

[6] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *ICCV*, 2019. 2

[7] Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fast convergence of DETR with spatially modulated co-attention. *CoRR*, abs/2101.07448, 2021. 2, 3

[8] Zigang Geng, Ke Sun, Bin Xiao, Zhaoxiang Zhang, and Jingdong Wang. Bottom-up human pose estimation via disentangled keypoint regression. In *CVPR*, pages 14676–14686, June 2021. 8

[9] Ross B. Girshick. Fast R-CNN. In *ICCV*, 2015. 2

[10] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010. 6

[11] Maosheng Guo, Yu Zhang, and Ting Liu. Gaussian transformer: A lightweight approach for natural language inference. In *AAAI*, 2019. 3

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 7

[13] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. In *NeurIPS*, 2018. 2

[14] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018. 2

[15] Lichao Huang, Yi Yang, Yafeng Deng, and Yinan Yu. Densebox: Unifying landmark localization with end to end object detection. *CoRR*, abs/1509.04874, 2015. 2

[16] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc Van Gool. Dynamic filter networks. In *NeurIPS*, 2016. 2

[17] Guolin Ke, Di He, and Tie-Yan Liu. Rethinking positional encoding in language pre-training. *CoRR*, abs/2006.15595, 2020. 3

[18] Jaeyoung Kim, Mostafa El-Khamy, and Jungwon Lee. T-GSA: transformer with gaussian-weighted self-attention for speech enhancement. In *ICASSP*, 2020. 3

[19] Tao Kong, Fuchun Sun, Huaping Liu, Yuning Jiang, and Jianbo Shi. Foveabox: Beyond anchor-based object detector. *CoRR*, abs/1904.03797, 2019. 2

[20] Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 1995. 6

[21] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, 2018. 2

[22] Hei Law, Yun Teng, Olga Russakovsky, and Jia Deng. Cornernet-lite: Efficient keypoint based object detection. In *BMVC*. BMVA Press, 2020. 2

[23] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. In *ICCV*, pages 6054–6063, 2019. 2

[24] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *TPAMI*, 2020. 2, 6

[25] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, 2014. 6

[26] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. In *ECCV*, 2016. 2

[27] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. In *ICLR*, 2017. 6

[28] Xin Lu, Buyu Li, Yuxin Yue, Quanquan Li, and Junjie Yan. Grid R-CNN. In *CVPR*, 2019. 2

[29] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra R-CNN: towards balanced learning for object detection. In *CVPR*, 2019. 2

[30] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 2

[31] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *CVPR*, 2017. 2

[32] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018. 2

[33] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *TPAMI*, 2017. 7

[34] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian D. Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, 2019. 6

[35] Guanglu Song, Yu Liu, and Xiaogang Wang. Revisiting the sibling head in object detector. In *CVPR*, 2020. 2

[36] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, pages 5693–5703, 2019. 8

[37] Zhiqing Sun, Shengcao Cao, Yiming Yang, and Kris Kitani. Rethinking transformer-based set prediction for object detection. *CoRR*, abs/2011.10881, 2020. 2, 7

[38] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *ECCV*, 2020. 2

[39] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: fully convolutional one-stage object detection. In *ICCV*, 2019. 2, 7

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2

[41] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui

参考文献

[1] Alexey Bochkovskiy, Chien-Yao Wang, 与 Hong-Yuan Mark Liao. YOLOv4：目标检测的最佳速度与精度. *CoRR*, abs/2004.10934, 2020. 2 [2] Zhaowei Cai 与 Nuno Vasconcelos. Cascade R-CNN：深入高质量目标检测. 载于 *CVPR*, 2018. 2 [3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, 及 Sergey Zagoruyko. 基于Transformer的端到端目标检测. 载于 *ECCV*, 2020. 1, 2, 3, 4, 5, 6, 7 [4] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, 与 Zicheng Liu. 动态卷积：卷积核上的注意力机制. 载于 *CVPR*, 2020. 2 [5] Zhigang Dai, Bolun Cai, Yugeng Lin, 与 Junying Chen. UP-DETR：基于Transformer的目标检测无监督预训练. *CoRR*, abs/2011.09094, 2020. 6, 7 [6] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, 与 Qi Tian. CenterNet：基于关键点三元组的目标检测. 载于 *ICCV*, 2019. 2 [7] Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, 与 Hongsheng Li. 空间调制协同注意力加速DETR收敛. *CoRR*, abs/2101.07448, 2021. 2, 3 [8] Zigang Geng, Ke Sun, Bin Xiao, Zhaoxiang Zhang, 与 Jingdong Wang. 基于解耦关键点回归的自底向上人体姿态估计. 载于 *CVPR*, 页14676–14686, 2021年6月. 8 [9] Ross B. Girshick. Fast R-CNN. 载于 *ICCV*, 2015. 2 [10] Xavier Glorot 与 Yoshua Bengio. 理解深度前馈神经网络训练难点. 载于 *AIS- TATS*, 2010. 6 [11] Maosheng Guo, Yu Zhang, 与 Ting Liu. 高斯Transformer：轻量级自然语言推理方法. 载于 *AAAI*, 2019. 3 [12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, 与 Jian Sun. 深度残差学习在图像识别中的应用. 载于 *CVPR*, 2016. 7 [13] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, 与 Andrea Vedaldi. Gather-Excite：利用卷积神经网络中的特征上下文. 载于 *NeurIPS*, 2018. 2 [14] Jie Hu, Li Shen, 与 Gang Sun. 压缩激励网络. 载于 *CVPR*, 2018. 2 [15] Lichao Huang, Yi Yang, Yafeng Deng, 与 Yinan Yu. DenseBox：端到端目标检测与地标定位的统一框架. *CoRR*, abs/1509.04874, 2015. 2 [16] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, 与 Luc Van Gool. 动态滤波器网络. 载于 *NeurIPS*, 2016. 2 [17] Guolin Ke, Di He, 与 Tie-Yan Liu. 重新思考语言预训练中的位置编码. *CoRR*, abs/2006.15595, 2020. 3 [18] Jaeyoung Kim, Mostafa El-Khamy, 与 Jungwon Lee. T-GSA：带高斯加权自注意力的语音增强Transformer. 载于 *ICASSP*, 2020. 3 [19] Tao Kong, Fuchun Sun, Huaping Liu, Yuning Jiang, 与 Jianbo Shi. FoveaBox：超越锚框的目标检测器. *CoRR*, abs/1904.03797, 2019. 2 [20] Harold W. Kuhn. 匈牙利算法求解分配问题. *Naval Research Logistics Quarterly*, 1995. 6

[21] 何毅、邓嘉。CornerNet：通过成对关键点检测目标。收录于*ECCV*，2018年。2 [22] 何毅、滕云、Olga Russakovsky、邓嘉。CornerNet-Lite：基于关键点的高效目标检测。收录于*BMVC*。BMVA出版社，2020年。2 [23] 李阳浩、陈云涛、王乃岩、张兆翔。尺度感知的三叉戟网络用于目标检测。收录于*ICCV*，第6054–6063页，2019年。2 [24] 林倞毅、Piya Goyal、Ross B. Girshick、何恺明、Piotr Dollár。密集目标检测的焦点损失。*TPAMI*，2020年。2, 6 [25] 林倞毅、Michael Maire、Serge J. Belongie、James Hays、Pietro Perona、Deva Ramanan、Piotr Dollár、C. Lawrence Zitnick。Microsoft COCO：上下文中的常见物体。收录于*ECCV*，2014年。6 [26] 刘伟、Dragomir Anguelov、Dumitru Erhan、Christian Szegedy、Scott E. Reed、傅成阳、Alexander C. Berg。SSD：单次多框检测器。收录于*ECCV*，2016年。2 [27] Ilya Loshchilov、Frank Hutter。修复Adam中的权重衰减正则化问题。收录于*ICLR*，2017年。6 [28] 陆昕、李步宇、岳雨欣、李全全、颜俊杰。Grid R-CNN。收录于*CVPR*，2019年。2 [29] 庞江淼、陈凯、史建平、冯华君、欧阳万里、林达华。Libra R-CNN：迈向均衡学习的目标检测。收录于*CVPR*，2019年。2 [30] Joseph Redmon、Santosh Kumar Divvala、Ross B. Girshick、Ali Farhadi。YOLO：统一实时的目标检测。收录于*CVPR*，2016年。2 [31] Joseph Redmon、Ali Farhadi。YOLO9000：更好、更快、更强。收录于*CVPR*，2017年。2 [32] Joseph Redmon、Ali Farhadi。YOLOv3：渐进式改进。*CoRR*，abs/1804.02767，2018年。2 [33] 任少卿、何恺明、Ross B. Girshick、孙剑。Faster R-CNN：基于区域提议网络的实时目标检测。*TPAMI*，2017年。7 [34] Hamid Rezatofighi、Nathan Tsoi、JungYoung Gwak、Amir Sadeghian、Ian D. Reid、Silvio Savarese。广义交并比：边界框回归的度量与损失函数。收录于*CVPR*，2019年。6 [35] 宋广鲁、刘宇、王晓刚。重新审视目标检测器中的兄弟头结构。收录于*CVPR*，2020年。2 [36] 孙科、肖斌、刘东、王井东。用于人体姿态估计的深度高分辨率表示学习。收录于*CVPR*，第5693–5703页，2019年。8 [37] 孙子庆、曹圣操、杨一鸣、Kris Kitani。重新思考基于Transformer的集合预测在目标检测中的应用。*CoRR*，abs/2011.10881，2020年。2, 7 [38] 田志、沈春华、陈浩。用于实例分割的条件卷积。收录于*ECCV*，2020年。2 [39] 田志、沈春华、陈浩、何通。FCOS：全卷积单阶段目标检测。收录于*ICCV*，2019年。2, 7 [40] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N. Gomez、Lukasz Kaiser、Illia Polosukhin。注意力机制就是你所需要的一切。收录于*NeurIPS*，2017年。2 [41] 王井东、孙科、程天恒、姜博瑞、邓超瑞、赵阳、刘东、慕亚东、Mingkui

Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *TPAMI*, 2019. 8

[42] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. Solov2: Dynamic and fast instance segmentation. In *NeurIPS*, 2020. 2

[43] Yifan Xu, Weijian Xu, David Cheung, and Zhuowen Tu. Line segment detection using transformers without edges. In *CVPR*, pages 4257–4266, June 2021. 8

[44] Brandon Yang, Gabriel Bender, Quoc V. Le, and Jiquan Ngiam. Condconv: Conditionally parameterized convolutions for efficient inference. In *NeurIPS*, 2019. 2

[45] Changqian Yu, Bin Xiao, Changxin Gao, Lu Yuan, Lei Zhang, Nong Sang, and Jingdong Wang. Lite-hrnet: A lightweight high-resolution network. In *CVPR*, pages 10440–10450, June 2021. 2

[46] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas S. Huang. Unitbox: An advanced object detection network. In *MM*, 2016. 2

[47] 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. 2

[48] Minghang Zheng, Peng Gao, Xiaogang Wang, Hongsheng Li, and Hao Dong. End-to-end object detection with adaptive clustering transformer. *CoRR*, abs/2011.09315, 2020. 2

[49] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *CoRR*, abs/1904.07850, 2019. 2

[50] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krähenbühl. Bottom-up object detection by grouping extreme and center points. In *CVPR*, 2019. 2

[51] Chenchen Zhu, Fangyi Chen, Zhiqiang Shen, and Marios Savvides. Soft anchor-point object detection. In *ECCV*, 2020. 2

[52] Chenchen Zhu, Yihui He, and Marios Savvides. Feature selective anchor-free module for single-shot object detection. In *CVPR*, 2019. 2

[53] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: deformable transformers for end-to-end object detection. *CoRR*, abs/2010.04159, 2020. 1, 2, 6, 7, 8

谭、王兴刚、刘文宇和肖斌。面向视觉识别的高分辨率深度表示学习。*TPAMI*，2019年。8 [42] 王新龙、张如峰、孔涛、李磊和沈春华。SOLOv2：动态快速的实例分割。收录于*NeurIPS*，2020年。2 [43] 徐一凡、徐伟健、张大卫和涂卓文。无需边缘的Transformer线段检测。收录于*CVPR*，第4257–4266页，2021年6月。8 [44] 杨布兰登、本德加布里埃尔、乐魁克和吴继全。条件参数化卷积的高效推理CondConv。收录于*NeurIPS*，2019年。2 [45] 于长谦、肖斌、高长新、袁璐、张磊、桑农和王京东。轻量级高分辨率网络Lite-HRNet。收录于*CVPR*，第10440–10450页，2021年6月。2 [46] 于嘉慧、姜宇宁、王章阳、曹志敏和黄汤姆。UnitBox：先进的目标检测网络。收录于*MM*，2016年。2 [47] 张世峰、迟程、姚永强、雷震和李斯坦。通过自适应训练样本选择弥合锚基与无锚检测的差距。收录于*CVPR*，2020年。2 [48] 郑明航、高鹏、王晓刚、李洪生和董浩。基于自适应聚类Transformer的端到端目标检测。*CoRR*，abs/2011.09315，2020年。2 [49] 周星艺、王德全和Philipp Krähenbūhl。以点代物。*CoRR*，abs/1904.07850，2019年。2 [50] 周星艺、朱家成和Philipp Krähenbūhl。通过极值点与中心点分组的自底向上目标检测。收录于*CVPR*，2019年。2 [51] 朱晨晨、陈方毅、沈志强和Marios Savvides。软锚点目标检测。收录于*ECCV*，2020年。2 [52] 朱晨晨、何一辉和Marios Savvides。单次目标检测中的特征选择无锚模块。收录于*CVPR*，2019年。2 [53] 朱希舟、苏伟杰、卢乐威、李斌、王晓刚和戴继峰。可变形DETR：端到端目标检测的可变形Transformer。*CoRR*，abs/2010.04159，2020年。1, 2, 6, 7, 8