DETRs与协作混合分配训练

宗卓凡 宋光禄 刘宇
商汤研究院
{zongzhuofan,liuyuisanai}@gmail.com
songguanglu@sensetime.com

摘要
本文提出了DETR中一对一集合匹配中正样本指定的查询过少导致编码器输出上的稀疏监督，从而严重损害了编码器的判别特征学习，反之，对于解码器中的注意力学习也是如此。为了缓解这个问题，我们提出了一种新的协作混合分配训练方案，即Co-DETR，从多种标签分配方式中学习更高效、更有效的DETR-based检测器。这种新的训练方案可以通过训练多个并行的辅助头部并使用一对多的标签分配（如ATSS和Faster RCNN）进行监督，从而轻松提升端到端检测器中编码器的学习能力。此外，我们通过从这些辅助头部提取正样本的坐标来提高解码器中正样本的训练效率。在推理阶段，这些辅助头部将被舍弃，因此我们的方法对原始检测器没有引入额外的参数和计算成本，同时也不需要手工制作非最大化抑制（NMS）。我们进行了大量实验证明了所提方法在包括ATSS在内的DETR变体上的有效性。

https://github.com/Sense-X/Co-DETR.

Swin-L 可以在 COCO 验证集上将平均准确率 (AP) 从 58.5% 提升至 59.5%。令人惊讶的是，搭配 ViT-L 骨干网络，我们在 COCO 测试集上实现了 66.0% 的 AP，LVIS 验证集上实现了 67.9% 的 AP，相比之前的方法有明显的提升，且模型尺寸更小。代码可在 https://github.com/Sense-X/Co-DETR 上获取。

1. 引言
   目标检测是计算机视觉中的一项基本任务，要求我们定位对象并对其进行分类。开创性的 R-CNN 系列算法 [11, 14, 27] 和一系列改进算法 [31, 37, 44]，如 ATSS [41]、RetinaNet [21]、FCOS [32] 和 PAA [17]，取得了目标检测任务的重大突破。它们的核心方案是一对多的标签分配，即将每个真实框分配给检测器输出中的多个坐标作为监督目标，与提议 [11, 27]、锚点 [21] 或窗口中心 [32] 协作。尽管它们的性能有所改善，但是这些检测器在很大程度上依赖于许多手工设计的组件，如非最大值抑制过程或锚点生成 [1]。为了构建一个更灵活的端到端检测器，我们提出了一种名为 DEtection TRansformer (DETR) 的方法 [1]，用于视图目标检测。

IoB 0.20.40.60.81.0 IoF

基于Transformer编码器-解码器结构的匹配方案。通过这种方式，每个真实框只会分配给一个特定的查询，不再需要多个手动设计的组件来编码先验知识。这种方法引入了一种灵活的检测流程，并鼓励许多DETR变体进一步改进它。然而，与一对多标签分配的传统检测器相比，原始端到端目标检测器的性能仍然较差。

在本文中，我们试图使基于DETR的检测器在保持其端到端优势的同时超越传统的检测器。为了应对这一挑战，我们关注一对一集合匹配的直观缺点，即它探索较少的正查询。这将导致严重的低效训练问题。我们从两个方面详细分析了这个问题，即由编码器生成的潜在表示和解码器中的注意力学习。我们首先比较了Deformable-DETR [43]和一对多标签分配之间的潜在特征的可辨别性得分，后者在训练过程中引入了额外的开销。

坐标轴用于表示可辨别性得分。给定编码器的输出F ∈RC×H×W，我们可以得到可辨别性得分图S ∈R1×H×W。当对应区域的得分较高时，可以更好地检测到对象。如图2所示，我们通过在可辨别性得分上应用不同的阈值来展示IoF-IoB曲线（IoF：前景交集率，IoB：背景交集率）（详见第3.4节）。ATSS中更高的IoF-IoB曲线表示更容易区分前景和背景。我们进一步在图3中可视化了可辨别性得分图S。很明显，一对多标签分配方法中的一些显著区域的特征被充分激活，但在一对一集合匹配中探索较少。为了对解码器训练进行探索，我们还展示了基于Deformable-DETR和Group-DETR [5]中的解码器中交叉注意力得分的IoF-IoB曲线，该方法引入了更多正查询。图2中的说明表明，正查询过少也会影响注意力学习，增加解码器中更多的正查询可以稍稍缓解这个问题。

这一重要观察启发我们提出了一种简单但有效的方法，即协作式混合分配训练方案（Co-DETR）。Co-DETR的关键观点是使用多功能的一对多标签分配来

编码器和解码器。更具体地，我们将辅助头部与变换器编码器的输出集成在一起。这些头部可以由多样化的一对多的标签分配进行监督，例如ATSS [41]，FCOS [32]和Faster RCNN [27]。不同的标签分配丰富了编码器输出上的监督，迫使其具备足够的区分度来支持这些头部的训练收敛。为了进一步提高解码器的训练效率，我们精确地编码了这些辅助头部中正样本的坐标，包括正锚点和正提议的坐标。它们被发送到原始解码器作为多个组的正查询，用于预测预分配的类别和边界框。每个辅助头部中的正坐标作为一个独立的组与其他组隔离。多样的一对多标签分配可以引入丰富的(正查询，真值)对来提高解码器的训练效率。值得注意的是，在推断过程中只使用原始解码器，因此提出的训练方案只在训练过程中引入额外的开销。
我们进行了大量实验证明所提方法的效率和有效性。如图3所示，Co-DETR在COCO验证集上将平均精度(average precision)从58.5%提升到59.5%。

插拔式方法：我们可以轻松地将其与不同的DETR变体结合使用，包括DAB-DETR [23]，Deformable-DETR [43]和DINO-Deformable-DETR [39]。如图1所示，Co-DETR实现了更快的训练收敛和更高的性能。具体而言，在12个周期的训练中，我们将基本的Deformable-DETR的AP提高了5.8%，在36个周期的训练中提高了3.2%。基于Swin-L [25]，目前最先进的DINO-Deformable-DETR在COCO val上的AP仍然可以从58.5%提高到59.5%。令人惊讶的是，结合ViT-L [8]骨干网络，我们在COCO test-dev上实现了66.0%的AP，在LVIS val上实现了67.9%的AP，建立了新的性能最好的检测器，并且模型大小大大减少。

2. 相关工作
   一对多的标签分配。在目标检测中，对于一对多的标签分配，可以将多个候选框分配给训练阶段的同一个真实框作为正样本。在经典的基于锚点的检测器中，例如Faster-RCNN [27]和RetinaNet [21]，样本选择是由预定义的IoU阈值和锚点与注释框之间的匹配IoU来指导的。基于中心先验的无锚点FCOS [32]利用中心先验和附近的空间位置分配每个边界框。
3. 方法
   3.1. 概述
   P{pos}i表示与真实框相匹配的正样本位置，

自适应锚点选择通过顶部k个最近锚点的统计动态IoU值。PAA [17]以概率方式自适应地将锚点分成正样本和负样本。在本文中，我们提出了一种协作混合分配方案，通过辅助头部进行一对多标签分配来改进编码器表示。

一对一集合匹配。开创性的基于Transformer的检测器DETR [1]将一对一集合匹配方案纳入目标检测中，并进行完全端到端的目标检测。一对一集合匹配策略首先通过匈牙利匹配计算全局匹配成本，并为每个真实框分配与最小匹配成本相对应的一个正样本。DN-DETR [18]展示了一对一集合匹配的不稳定性导致收敛速度慢的结果，因此引入去噪训练来消除这个问题。DINO [39]继承了DAB-DETR [23]的先进查询公式，并结合了一种改进的对比去噪技术，从而实现了最先进的性能。Group-DETR [5]构建了基于组的一对多标签分配，以利用多个正对象查询，类似于H-DETR [16]中的混合匹配方案。与以上后续工作相比，我们提出了一种新的协作优化的一对一集合匹配视角。

3. 方法
   3.1. 概述
   按照标准DETR协议，输入图像可以表示为：
   P{pos}
   i
   6

"并且通过协同混合分配训练方案和定制的正查询生成，在解码器中进行注意力学习。我们将详细描述这些模块，并解释为什么它们可以很好地发挥作用。
3.2. 协同混合分配训练
为了解决解码器中正查询更少导致的对编码器输出稀疏监督的问题，我们将具有不同一对多标签分配范式（例如ATSS和Faster R-CNN）的多功能辅助头部进行整合。不同的标签分配丰富了对编码器输出的监

督，迫使其具有足够的区分度以支持这些头部的训练收敛。具体而言，给定编码器的潜在特征F，我们首先通过多尺度适配器将其转换为特征金字塔{F1，…，FJ}，其中J表示具有22+J下采样步幅的特征图。类似于ViTDet[20]，特征金字塔由单个特征图在单尺度编码器中构建，而我们使用双线性插值和3×3卷积进行上采样。例如，对于来自编码器的单尺度特征，我们通过连续应用下采样（stride=2的3×3卷积）或上采样操作来生成特征金字塔。对于多尺度编码器，我们只对多尺度编码器特征F中最粗糙的特征进行下采样，以构建特征金字塔。为第i个i定义了K个协同头部和相应的标签分配方式Ak的情况。"

预测结果ˆPi。在第i个头部，Ai被用来计算Pi中正样本和负样本的监督目标。将地面真值集合表示为Gas，这个过程可以表示为：
P{pos}i,B{pos}i,P{neg}i=Ai(ˆPi,G), (1)
其中{pos}和{neg}表示在Fj中由Ai确定的(j，正样本坐标或负样本坐标)的一对集合。Ai.j表示{F1,⋯,FJ}中的特征索引。B{pos}i 是
3Head i LossLiAssignment Ai
{pos},{neg}Generation PiGeneration B{pos}i Generation
Faster-RCNN [27]cls: CE损失，{pos}: IoU(proposal, gt) >0.5 {pos}: gt标签，偏移(proposal, gt)正样本提案
reg: GIoU损失 {neg}: IoU(proposal, gt) <0.5 {neg}: gt标签 (x1, y1, x2, y2)
ATSS [41]cls: Focal损失 {pos}:IoU(anchor, gt) >(mean+std) {pos}: gt标签，偏移(anchor, gt)，中心度正锚点
reg: GIoU，BCE损失 {neg}: IoU(anchor, gt) <(mean+std) {neg}: gt标签 (x1, y1, x2, y2)
RetinaNet [21]cls: Focal损失 {pos}: IoU(anchor, gt) >0.5 {pos}: gt标签，偏移(anchor, gt)正锚点
reg: GIoU损失 {neg}: IoU(anchor, gt) <0.4 {neg}: gt标签 (x1, y1, x2, y2)
FCOS [32]cls: Focal损失 {pos}: 在gt中心区域的点 {pos}: gt标签，ltrb距离，中心度FCOS点(cx, cy)
reg: GIoU，BCE损失 {neg}: 不在gt中心区域的点 {neg}: gt标签 w=h= 8×22+j
表1. 辅助头部的详细信息。辅助头部包括Faster-RCNN [27]，ATSS [41]，RetinaNet [21]和空间正坐标集合。P{pos}i和P{neg}i
Lenc PH

包括类别和回归偏移量。具体来说，我们在表1中描述了每个变量的详细信息。损失函数可以定义为：
L*{enc}*=*sum*{i=1}^{N}[L_i(\hat{P}*{\text{pos},i},P*{\text{pos},i})+L_i(\hat{P}*{\text{neg},i},P*{\text{neg},i})] \quad (2)
请注意，对于负样本，回归损失被丢弃。K补充头部的优化训练目标如下所示：
L*{enc}*=*sum*{i=1}^{K}L*{enc,i}* *quad (3)*
3.3. 定制的正查询生成
在一对一的集合匹配范式中，每个真实框只会被分配给一个特定的查询作为监督目标。正查询太少会导致Transformer解码器中的跨注意力学习低效，如图2所示。为了缓解这个问题，我们根据每个辅助头的标签分配A_i，精心生成足够的定制正查询。具体地，对于第i个辅助头中的正样本坐标集合
B{\text{pos},i} \in \mathbb{R}^{M_i \times 4}，其中M_i是正样本数量，可以通过以下方式生成额外的定制正查询Q_i \in \mathbb{R}^{M_i \times C}：
Q_i = \text{Linear}(PE(B*{\text{pos},i})) + *text{Linear}(E(F^*,\text{pos})). \quad (4)*
这里，PE(\cdot)代表位置编码，并且根据索引对(j, F_j中的正坐标或负坐标)从E(\cdot)中选择相应的特征。
因此，有K+1组查询对单一的一对一集合匹配分支起到贡献，以及K个分支对应于训练期间的一对多标签分配。
L{dec,i,l} = -e^L(eP_i^l,P_{\text{pos},i}). \quad (5)
PH

被舍弃。具体而言，第i个辅助分支中第l个解码器层的损失可以表示为：
L*{\text{dec},i,l} = e \cdot L(eP*{\text{dec},i,l}, P*{\text{pos},i}). (5)*
其中eP{\text{dec},i,l}表示第i个辅助分支中第l个解码器层的输出预测。最后，Co-DETR的训练目标为：
L*{\text{global}}* = *sum*{l=1}^{L*{\text{dec}}} (e \cdot L*{\text{dec},l} + \lambda_1 \cdot \frac{K}{X} \sum*{i=1}^{X} L*{\text{dec},i,l} + \lambda_2 \cdot L*{\text{enc}}), (6)*
其中L{\text{dec},l}表示原始的一对一匹配分支的损失[1]，λ1和λ2是平衡损失的系数。

3.4. Co-DETR的工作原理
Co-DETR显著改进了基于DETR的检测器。接下来，我们将以定性和定量的方式研究其有效性。我们使用36个时期的设置，基于带有ResNet-50 [15]骨干网络的可变形DETR进行详细分析。
丰富编码器的监督。直观上，过少的正样本查询导致监督变得稀疏，因为每个真实样本只有一个查询通过回归损失进行监督。采用一对多标签分配方式的正样本接收更多的定位监督，以帮助增强潜在特征学习。为了进一步探索稀疏监督如何阻碍模型训练，我们详细地研究了编码器产生的潜在特征。我们引入了IoF-IoB曲线来量化编码器输出的可辨别性得分。具体来说，给定编码器的潜在特征F，受图3中特征可视化的启发，我们计算IoF（交集比）：
D(F) =1-\frac{1}{JX}\sum_{j=1}^{JX}bF_j

图像的尺寸为H×W。判别能力得分D(F)通过对所有层的得分进行平均计算：
D(F) =1/JJX j=1bFj/max(bFj), (7)
42 4 6 8 10 12
Epoch101112IS(Instability)
Deformable-DETR
Co-Deformable-DETR图5. Deformable-DETR和Co-Deformable-DETR在COCO数据集上的不稳定性（IS）[18]。这些检测器使用ResNet-50骨干网络进行12个时期的训练。
此处省略了调整大小操作。我们在图3中可视化了ATSS、Deformable-DETR和我们的Co-Deformable-DETR的判别能力得分。与Deformable-DETR相比，ATSS和Co-Deformable-DETR都具有更强的区分关键对象区域的能力，而Deformable-DETR几乎受到背景的干扰。因此，我们将前景和背景的指标定义为1(D(F)> S)∈RH×W和1(D(F)< S)∈RH×W，其中S是预定义的得分阈值，1(x)在x为真时为1，否则为0。对于前景的掩码Mfg∈RH×W，如果点(h, w)位于前景内，则元素Mfg
h,w为1，否则为0。前景的交集区域（IoF）Ifg可以计算如下：
Ifg=PH
h=1PW
w=1(1(D(Fh,w)> S)· Mfg
h,w)
PH
h=1PW
w=1Mfg
h,w.(8)
具体来说，我们以类似的方式计算背景的交集区域（IoB）的面积，并通过在图2中改变S来绘制IoF和IoB曲线。显然，在相同的IoB值下，ATSS和Co-Deformable-DETR的IoF值高于Deformable-DETR和Group-DETR，这证明了编码器表示+ PAA的优越性。

通过减少匈牙利匹配的不稳定性，提高交叉注意力学习。匈牙利匹配是一对一集合匹配中的核心方案。交叉注意力是一个重要的操作，可以帮助正查询编码丰富的对象信息。为了实现这一点，需要充分的训练。我们观察到，在训练过程中，匈牙利匹配引入了不可控的不稳定性，因为在同一图像中，分配给特定正查询的真实值在训练过程中会发生变化。根据[18]的结论，我们在图5中展示了不稳定性的比较，发现我们的方法有助于更稳定的匹配过程。此外，为了量化交叉注意力的优化程度，我们还计算了注意力分数的IoF-IoB曲线。与特征可区分性分数计算类似，我们为注意力分数设置不同的阈值，以获得多个IoF-IoB对。在图2中可以看到Deformable-DETR、Group-DETR和Co-Deformable-DETR之间的比较。我们发现，具有更多正查询的DETRs的IoF-IoB曲线通常高于Deformable-DETR，这与我们的动机一致。

3.5. 与其他方法的比较
我们的方法与其他方法的区别。Group-DETR、H-DETR和SQR [2]通过重复分组和重复的真实框进行一对多的分配。Co-DETR明确地将多个空间坐标分配为正查询。

- PAA

信号直接应用于潜在特征图，使其更具辨别力。相比之下，Group-DETR、H-DETR和SQR都缺乏这种机制。虽然这些对应方法引入了更多的正查询，但匈牙利匹配实现的一对多分配仍然存在一对一匹配的不稳定问题。我们的方法受益于现成的一对多分配的稳定性，并继承了它们在正查询和真实边界框之间特定匹配方式。Group-DETR和H-DETR未能揭示一对一匹配和传统一对多分配之间的互补性。据我们所知，我们是首次对具有传统一对多分配和一对一匹配的检测器进行定量和定性分析。这有助于我们更好地理解它们的差异和互补性，以便我们能够自然地通过利用现成的一对多分配设计来提高DETR的学习能力，而无需额外的专门的一对多设计经验。

解码器中没有引入负查询。重复的对象查询必然为解码器带来大量的负查询和显著增加的GPU内存消耗。然而，我们的方法只处理解码器中的正坐标，因此在表7中显示出占用更少的内存。

4.实验
4.1.设置
数据集和评估指标。我们的实验是在PA数据集上进行的。

v1.0数据集 [12]。COCO数据集包括115K张带标注的训练图像和5K张验证图像。我们默认在验证子集上报告检测结果。我们在5Method K＃epochs AP下报告了最大模型的评估结果。

条件DETR-C5 [26] 0 36 39.4
条件DETR-C5 [26] 1 36 41.5(+2.1)
条件DETR-C5 [26] 2 36 41.8(+2.4)
DAB-DETR-C5 [23] 0 36 41.2
DAB-DETR-C5 [23] 1 36 43.1(+1.9)
DAB-DETR-C5 [23] 2 36 43.5(+2.3)
可变-DETR [43] 0 12 37.1
可变-DETR [43] 1 12 42.3(+5.2)
可变-DETR [43] 2 12 42.9(+5.8)
可变-DETR [43] 0 36 43.3
可变-DETR [43] 1 36 46.8(+3.5)
可变-DETR [43] 2 36 46.5(+3.2)

表2. COCO验证集上普通基线的结果。还报告了test-dev（20K张图像）的结果。LVIS v1.0是一个具有1203个类别的大规模和长尾数据集，用于大词汇实例分割。为了验证Co-DETR的可扩展性，我们进一步将其应用于一个大规模的目标检测基准，即Objects365 [30]。在Objects365数据集中，用于训练的标记图像有1.7M张，验证图像有80K张。所有结果都遵循标准的平均精度（AP），具有不同目标尺度下IoU阈值从0.5到0.95的范围。

实施细节。我们将Co-DETR整合到当前的DETR-like管道中，并保持与基线的训练设置一致。我们采用ATSS和Faster-RCNN作为K= 2和paste [10]的辅助头部。

4.2. 主要结果

- PAA
  |D|X

在这一节中，我们通过对表2和表3中不同DETR变体的实证分析，来评估Co-DETR的有效性和泛化能力。所有的结果都是使用mmdetection [4]复现的。
我们首先对具有C5特征的单尺度DETR应用协同混合指派训练。令人惊讶的是，无论是Conditional-DETR还是DAB-DETR，都在较长的训练周期内相对于基线模型取得了2.4%和2.3%的准确率提升。对于具有多尺度特征的Deformable-DETR，检测性能从37.1%提升到了42.9%准确率。在将训练时间增加到36个时期后，整体改进仍然保持在+3.2%的准确率。此外，我们按照[16]对改进后的Deformable-DETR（称为Deformable-DETR++）进行了实验，结果显示准确率提高了2.4%。
在COCO验证集上的强基线结果如表3所示。方法得到了从58.5%到59.5%准确率的提升。

- PAA

我们进一步根据两个最先进的基准模型，将骨干网络容量从ResNet-50扩展到Swin-L [25]。如表3所示，Co-DETR实现了56.9%的AP，大幅超过了变形-DETR++基准的1.7% AP。使用Swin-L的DINO-变形-DETR的性能仍然可以从58.5%提升到59.5% AP。

4.3. 与最先进方法的比较

我们将K设置为2应用到变形-DETR++和DINO中。同时，我们采用了品质焦点损失[19]和非最大抑制（NMS）来优化我们的共同DINO-变形-DETR。我们在COCO验证集上进行比较，结果如表4所示。与其他竞争对手相比，我们的方法收敛速度更快。例如，仅使用ResNet-50骨干网络进行12个轮次训练，共同DINO-变形-DETR就能轻松达到52.1%的AP。我们使用Swin-L在1×调度器下可以获得58.9%的AP，甚至在3×调度器下超过其他最先进的框架。更重要的是，我们的最佳模型共同DINO-变形-DETR++在36个轮次的训练下实现了54.8%的AP（使用ResNet-50）和60.7%的AP（使用Swin-L），明显超过所有使用相同骨干网络的现有检测器。

为了进一步探索我们方法的可扩展性，我们将骨干网络容量扩展到3.04亿个参数。这个大规模骨干网络ViT-L [7]使用了自监督学习方法（EV A-02 [8]）进行预训练。我们首先在Objects365数据集上对ViT-L进行了26个轮次的预训练，然后在COCO数据集上进行微调，得到†：5个特征层的共同DINO-变形-DETR模型。

被随机选择的输入图像尺寸在480×2400和1536×2400之间。详细设置可以在补充材料中找到。我们的结果是通过测试时增强进行评估的。表5展示了对6 Method的最新比较结果。

\begin{center}
\begin{tabular}{ccccccccccc}
\hline
Method & Backbone & Multi-scale & #query & #epochs & AP & AP 50 & AP75 & APS & APM & APL \
\hline
Conditional-DETR [26] & R50 & ✗ & 300 & 108 & 43.0 & 64.0 & 45.7 & 22.7 & 46.7 & 61.5 \
Anchor-DETR [35] & R50 & ✗ & 300 & 50 & 42.1 & 63.1 & 44.9 & 22.3 & 46.2 & 60.0 \
DAB-DETR [23] & R50 & ✗ & 900 & 50 & 45.7 & 66.2 & 49.0 & 26.1 & 49.4 & 63.1 \
AdaMixer [9] & R50 & ✓ & 300 & 36 & 47.0 & 66.0 & 51.1 & 30.1 & 50.2 & 61.8 \
Deformable-DETR [43] & R50 & ✓ & 300 & 50 & 46.9 & 65.6 & 51.0 & 29.6 & 50.1 & 61.6 \
DN-Deformable-DETR [18] & R50 & ✓ & 300 & 50 & 48.6 & 67.4 & 52.7 & 31.0 & 52.0 & 63.7 \
DINO-Deformable-DETR†[39] & R50 & ✓ & 900 & 12 & 49.4 & 66.9 & 53.8 & 32.3 & 52.5 & 63.9 \
DINO-Deformable-DETR†[39] & R50 & ✓ & 900 & 36 & 51.2 & 69.0 & 55.8 & 35.0 & 54.3 & 65.3 \
DINO-Deformable-DETR†[39] & Swin-L (IN-22K) & ✓ & 900 & 36 & 58.5 & 77.0 & 64.1 & 41.5 & 62.3 & 74.0 \
Group-DINO-Deformable-DETR [5] & Swin-L (IN-22K) & ✓ & 900 & 36 & 58.4 & - & - & 41.0 & 62.5 & 73.9 \
H-Deformable-DETR [16] & R50 & ✓ & 300 & 12 & 48.7 & 66.4 & 52.9 & 31.2 & 51.5 & 63.5 \
H-Deformable-DETR [16] & Swin-L (IN-22K) & ✓ & 900 & 36 & 57.9 & 76.8 & 63.6 & 42.4 & 61.9 & 73.4 \
Co-Deformable-DETR & R50 & ✓ & 300 & 12 & 49.5 & 67.6 & 54.3 & 32.4 & 52.7 & 63.7 \
Co-Deformable-DETR & Swin-L (IN-22K) & ✓ & 900 & 36 & 58.5 & 77.1 & 64.5 & 42.4 & 62.4 & 74.0 \
Co-DINO-Deformable-DETR† & R50 & ✓ & 900 & 12 & 52.1 & 69.4 & 57.1 & 35.4 & 55.4 & 65.9 \
Co-DINO-Deformable-DETR† & Swin-L (IN-22K) & ✓ & 900 & 12 & 58.9 & 76.9 & 64.8 & 42.6 & 62.7 & 75.1 \
Co-DINO-Deformable-DETR† & Swin-L (IN-22K) & ✓ & 900 & 24 & 59.8 & 77.7 & 65.5 & 43.6 & 63.5 & 75.5 \
\hline
\end{tabular}
\end{center}

†表示5个特征级别。

Co-DINO-Deformable-DETR++†R50 ✓ 900 36 54.8 72.5 60.1 38.3 58.4 69.6
Co-DINO-Deformable-DETR++†Swin-L（IN-22K）✓ 900 12 59.3 77.3 64.9 43.3 63.3 75.5
Co-DINO-Deformable-DETR++†Swin-L（IN-22K）✓ 900 24 60.4 78.3 66.4 44.6 64.2 76.5
Co-DINO-Deformable-DETR++†Swin-L（IN-22K）✓ 900 36 60.7 78.5 66.7 45.1 64.7 76.4
†：5个特征层。
表4. 在COCO验证集上与最先进DETR变体的比较。
方法 骨干编码 验证集 测试-发展集
#params APboxAPbox
HTC++ [3] SwinV2-G [24] 3.0B 62.5 63.1
DINO [39] Swin-L [25] 218M 63.2 63.3
BEIT3 [33] ViT-g [7] 1.9B - 63.7
FD [36] SwinV2-G [24] 3.0B - 64.2
DINO [39] FocalNet-H [38] 746M 64.2 64.3
Group DETRv2 [6] ViT-H [7] 629M - 64.5
EV A-02 [8] ViT-L [7] 304M 64.1 64.5
DINO [39] InternImage-G [34] 3.0B 65.3 65.5
Co-DETR ViT-L [7] 304M 65.9 66.0
表5. 在COCO上与最先进框架的比较。
COCO测试-发展集基准。只使用304M参数的模型，Co-DETR在COCO测试-发展集上取得了66.0%的AP，超过了之前最佳模型InternImage-G [34] 的+0.5% AP。
我们还展示了Co-DETR在长尾LVIS检测数据集上的最佳结果。特别地，我们使用与COCO相同的Co-DINO-Deformable-DETR++模型，但选择FedLoss [42]作为分类损失以修复不平衡数据分布的影响。在这里，我们只应用边界框监督并报告目标检测结果。比较结果如下：
在LVIS验证集和最小验证集上达到了62.3%的AP，超过了ViT-

在LVIS验证集和最小验证集上，H-DETR [16]使用Swin-L [25]作为骨干网络，取得了62.3%的AP，超过了使用MAE预训练的ViT-H和GLIPv2 [40] [13]。ViTDet [20]使用ViT-L [7]作为骨干网络，获得了51.2%的AP，ViT-H [7]获得了632M的APbox和53.4%的AP。GLIPv2 [40]使用Swin-H [25]作为骨干网络，取得了637M的APbox和59.8%的AP。DINO [39]使用InternImage-G [34]作为骨干网络，达到了3.0B的APbox和63.2%的AP。EV A-02 [8]使用ViT-L [7]作为骨干网络，获得了304M的APbox和65.2%的AP。Co-DETR使用Swin-L [25]和ViT-L [7]作为骨干网络，分别取得了218M的APbox和56.9%的AP，以及304M的APbox和67.9%的AP。在LVIS验证集和最小验证集上，我们的方法在不进行复杂的测试时增强的情况下，获得了67.9%和71.9%的AP，表现最佳。与具有30亿参数的InternImage-G相比，在减小模型大小到原模型的1/10的同时，我们在LVIS验证集和最小验证集上分别获得了+4.7%和+6.1%的AP增益。

除非另有说明，所有消融实验都是在具有ResNet-50骨干网络的Deformable-DETR上进行的。我们将辅助头的数量K默认设置为1，并将总批量大小设置为32。更多的消融和实验结果请参见表格7。

表格7. 在LVIS上与最新框架的比较。

方法 骨干网络 辅助头 内存 GPU AP
(MB) 小时
Deformable-DETR++ 0 - 12808 70 47.1
H-Deformable-DETR 0 - 15307 104 48.4
Deformable-DETR++ 1 ATSS 13947 86 48.7
Deformable-DETR++ 2 ATSS + PAA 14629 124 49.0
Deformable-DETR++ 2 ATSS + Faster-RCNN 14387 120 49.5
Deformable-DETR++ 3 ATSS + Faster-RCNN + PAA 15263 150 49.5

使用Deformable-DETR++ 6ATSS + Faster-RCNN、19385 280 48.9 + PAA + RetinaNet、+ FCOS + GFL等方法进行了实验，结果如下表所示：

\begin{table}[h]
\centering
\begin{tabular}{cccccc}
\hline
K & Auxiliary head & #epochs & AP & AP50 & AP75 \
\hline
1 & Baseline & 36 & 43.3 & 62.3 & 47.1 \
1 & RetinaNet [21] & 36 & 46.1 & 64.2 & 50.1 \
1 & Faster-RCNN [27] & 36 & 46.3 & 64.7 & 50.5 \
1 & Mask-RCNN [14] & 36 & 46.5 & 65.0 & 50.6 \
1 & FCOS [32] & 36 & 46.5 & 64.8 & 50.7 \
1 & PAA [17] & 36 & 46.5 & 64.6 & 50.7 \
1 & GFL [19] & 36 & 46.5 & 65.0 & 51.0 \
1 & ATSS [41] & 36 & 46.8 & 65.1 & 51.5 \
\hline
\end{tabular}
\caption{K从1到6的实验结果}
\end{table}

在选择辅助头部的标准方面，我们在表7和8中进一步探讨了选择辅助头部的标准。从表8的结果可以看出，任何带有一对多标签分配的辅助头部都能够持续改善基准线的性能，而ATSS实现了最佳性能。我们发现，当选择K小于3时，准确度会随着K的增加而不断提高。值得注意的是，当K=6时，性能会下降，我们推测这是由于辅助头部之间存在严重冲突造成的。如果辅助头部的特征学习不一致，当K变大时，持续的改进将被破坏。我们还在接下来的章节和补充材料中分析了多个头部的优化一致性。总之，我们可以选择任何头部作为辅助头部，并且在K≤2时，ATSS和Faster-RCNN被认为是实现最佳性能的常见做法。我们不会优化冲突。

优化冲突。

冲突分析。当同一空间坐标被分配给不同的前景框或在不同的辅助头部中被视为背景时，会出现冲突，这可能导致检测器的训练混乱。我们首先定义头部Hi和头部Hj之间的距离，以及Hito的平均距离来衡量优化冲突为：

[
S*{i,j}=\frac{1}{|D|} \sum*{{I \in D}} KL(C(H_i(I)),C(H_j(I))), (9)
]

ATSS Faster-RCNN PAA GFL FCOS RetinaNet0.000.010.02距离K=1
K=2
K=3
K=6Figure 6. 当K从1变化到6时的距离。

辅助头部 位置查询数 #epochs AP AP50 AP75
✗ ✗12 37.1 55.5 40.0
36 43.3 62.3 47.1
✓ ✗12 41.6(+4.5) 59.8 45.6
36 46.2(+2.9) 64.7 50.9
✗ ✓12 40.5(+3.4) 58.8 44.4
36 45.3(+2.0) 63.5 49.8
✓ ✓12 42.3(+5.2) 60.5 46.1
36 46.8(+3.5) 65.1 51.5
表9。“aux head”表示使用辅助头部进行训练，“pos queries”表示自定义正查询生成。

[
S_i=\frac{1}{2(K-1)K} \sum{{j \neq i}} (S{i,j}+S_{j,i}), (10)
]

其中KL，D，I，C分别表示KL散度，数据集，输入图像和类激活映射（CAM）[29]。如图6所示，我们计算K>1时辅助头部之间的平均距离，以及K=1时DETR头部与单个辅助头部之间的距离。我们发现，当K=1时，每个辅助头部的距离度量是不显著的，这与表8中的结果一致：当K=1时，DETR头部可以与任何头部协同改进。当K增加到2时，辅助头部之间的距离度量变得更加重要。

ATSS实现了49.5%的AP，并且可以通过将ATSS替换为6个不同的辅助头将其降至48.9%的AP。因此，我们推测过多的不同辅助头，例如超过3个不同的头部，会加剧冲突。总结起来，优化冲突受到不同辅助头的数量和这些头部之间的关系的影响。添加的头部应该是不同的吗？与一个ATSS头（48.7% AP）相比，与两个ATSS头的协同训练（49.2% AP）仍然会改善模型，因为根据我们的分析，ATSS对DETR头起到了补充作用。此外，引入一个多样且互补的辅助头而不是与原始头部相同的辅助头部，例如Faster-RCNN，可以带来更好的性能提升（49.5% AP）。注意，这并不与上述结论相矛盾；相反，我们可以通过少量不同的头部（K≤2）获得最佳性能，因为冲突较小，但如果使用太多不同的头部（K > 3），我们将面临严重的冲突。

\begin{table}[h]
\centering
\caption{与更长训练周期的基准模型比较}
\begin{tabular}{cccc}
\textbf{Method} & \textbf{K} & \textbf{#epochs} & \textbf{GPU hours} & \textbf{AP} \
\hline
Deformable-DETR & 1 & 36 & 288 & 46.8 \
Deformable-DETR & 0 & 50 & 333 & 44.5 \
Deformable-DETR & 0 & 100 & 667 & 46.0 \
Deformable-DETR & 0 & 150 & 1000 & 45.9 \
\end{tabular}
\end{table}

\begin{table}[h]
\centering
\caption{使用不同数量的辅助头的协同训练对Deformable-DETR++ with ResNet-50各分支性能的持续改进}
\begin{tabular}{cccc}
\textbf{Branch} & \textbf{NMS} & \textbf{K=0} & \textbf{K=1} & \textbf{K=2} \
\hline
Deformable-DETR++ & \xmark & 47.1 & 48.7 (+1.6) & 49.5 (+2.4) \
ATSS & \checkmark & 46.8 & 47.4 (+0.6) & 48.0 (+1.2) \
Faster-RCNN & \checkmark & 45.9 & - & 46.7 (+0.8) \
\end{tabular}
\end{table}

引入一个多样且互补的辅助头，而不是相同的原始头部，例如Faster-RCNN，可以带来更好的收益（49.5% AP）。请注意，这并不与上述结论相矛盾；相反，当冲突较小时，我们可以通过少量不同的头部（K≤2）获得最佳性能，但是当使用许多不同的头部（K > 3）时，我们面临严重的冲突。

有效的注意力学习对于解码器非常重要。

参考文献

Table 9中的结果表明，引入辅助头部可显著提高性能，因为密集的空间监督使得编码器特征更具辨别力。另外，引入定制的正样本查询也对最终结果有明显贡献，同时提高了一对一集匹配的训练效率。这两种技术都可以加快收敛并提高性能。总体而言，我们观察到整体的改进来自于编码器的更具辨别力的特征和解码器的更高效的注意力学习。

与较长的训练计划进行比较。如表10所示，我们发现变形DETR不能从较长的训练中获益，因为性能饱和。相反，Co-DETR大大加快了收敛速度，并提高了性能峰值。

辅助分支的性能。令人惊讶的是，我们观察到Co-DETR在辅助头部的表现也有一致的提升（见表11）。这意味着我们的训练模式有助于产生更具辨别力的编码器表示，从而改善了解码器和辅助头部的性能。

原始正样本查询和定制正样本查询分布的差异。我们在图7a中可视化了原始正样本查询和定制正样本查询的位置。我们每张图片只显示一个对象（绿色框）。解码器使用匈牙利匹配分配的正样本查询用红色标记。我们标记了从...

5. 结论

这些定制化查询被分布在实例的中心区域，并为检测器提供足够的监督信号。

分布差异是否会导致不稳定性？我们在图7b中计算了原始查询和定制查询之间的平均距离。原始负查询与定制正查询之间的平均距离显著大于原始查询和定制正查询之间的距离。由于原始查询和定制查询之间的分布差距很小，因此在训练过程中没有遇到不稳定性。

结论
在本文中，我们提出了一种新的协作混合分配训练方案，称为Co-DETR，可以从多样的标签分配方式中学习更高效和有效的基于DETR的检测器。这种新的训练方案可以通过训练多个并行的辅助头，这些头以一对多的标签分配方式为监督，轻松提升编码器的学习能力，并通过从这些辅助头中提取正样本的坐标来进行额外的定制正查询，以提高解码器中正样本的训练效率。在COCO数据集上进行了大量实验证明了Co-DETR的效率和有效性，在COCO测试集上达到了66.0%的平均精确度。

在LVIS验证集上，建立了新的最先进的检测器，使用更少的模型大小。

参考文献
[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. ArXiv, abs/2005.12872, 2020. 1, 3, 4
[2] Fangyi Chen, Han Zhang, Kai Hu, Yu-kai Huang, Chenchen Zhu, and Marios Savvides. Enhanced training of query-based object detection via selective query recollection. arXiv preprint arXiv:2212.07593, 2022. 5
[3] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, 等。负责任的任务级联实例分割。在计算机视觉和模式识别的IEEE/CVF会议论文集中的论文，第4974-4983页，2019年。 7
[4] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, 等。Mmdetection:开放mmlab检测工具箱和基准。arXiv预印本arXiv:1906.07155，2019年。 6
[5] Qiang Chen, Xiaokang Chen, Gang Zeng, and Jingdong Wang. Group detr：通过解耦的一对多标签分配实现快速训练收敛。arXiv预印本arXiv:2207.13085，2022年。 2, 3, 7
[6] Qiang Chen, Jian Wang, Chuchu Han, Shan Zhang, Zexian Li, Xiaokang Chen, Jiahui Chen, Xiaodi Wang, Shuming Han, Gang Zhang, 等。Group detr v2:带有编码器-解码器预训练的强大对象检测器。arXiv预印本arXiv:2211.03594，2022年。 6, 7
[7] 7

Mostafa Dehghani等人在2021年的文章《图像等于16x16个单词：大规模图像识别的Transformer》中介绍了一种用于图像识别的Transformer模型。这篇文章发表在ArXiv上，编号为abs/2010.11929。Yuxin Fang等人在2023年的文章《eva-02：新世纪福音战士的视觉表现》中提出了一种新的视觉表现模型，这篇文章发表在arXiv上，编号为arXiv:2303.11331。Ziteng Gao等人在2022年的文章《Adamixer：一种收敛速度快的基于查询的对象检测器》中介绍了一种快速收敛的对象检测方法。这篇文章发表在IEEE/CVF计算机视觉与模式识别会议上，页码为5364-5373。Golnaz Ghiasi等人在2021年的文章《简单的复制粘贴是一种强大的实例分割数据增强方法》中介绍了一种简单的复制粘贴数据增强方法，对实例分割任务效果显著。这篇文章发表在IEEE/CVF计算机视觉与模式识别会议上，页码为2918-2928。Ross Girshick在2015年的文章《Fast R-CNN》中提出了一种快速的R-CNN算法。这篇文章发表

在IEEE国际计算机视觉会议上，页码为1440-1448。Agrim Gupta等人在2019年的文章《LVIS：一个用于大词汇实例分割的数据集》中介绍了一个用于大词汇实例分割的数据集。这篇文章发表在IEEE/CVF计算机视觉与模式识别会议上，页码为5356-5364。Kaiming He等人在2022年的文章《Masked Autoencoders是可扩展的视觉学习器》中介绍了一种可扩展的视觉学习模型。这篇文章发表在IEEE/CVF计算机视觉与模式识别会议上，页码为16009。

尊敬的读者：

我们将以下内容翻译为中文：

[15] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. 深度残差学习用于图像识别. 2016年IEEE计算机视觉和模式识别会议(CVPR), 2016, 页码770-778. 第4条引用。

[16] Ding Jia, Yuhui Yuan, Haodi He, Xiaopei Wu, Haojun Yu, Weihong Lin, Lei Sun, Chao Zhang, 和 Han Hu. 使用混合匹配的Detrs. arXiv预印本arXiv:2207.13080, 2022, 页码3, 6, 7, 12.

[17] Kang Kim 和 Hee Seok Lee. 基于IoU预测的概率锚点分配用于目标检测. 在欧洲计算机视觉会议(ECCV), 页码355-371, Springer, 2020, 页码1, 3, 8, 13.

[18] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, 和 Lei Zhang. DN-DETR: 引入查询去噪以加速DETR训练. 在IEEE/CVF计算机视觉和模式识别会议(CVPR), 页码13619-13627, 2022, 页码3, 5, 7.

[19] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, 和 Jian Yang. 广义聚焦损失: 学习适用于密集目标检测的合格和分布式边界框. 在神经信息处理系统(Advances in Neural Information Processing Systems), 第33卷: 21002-21012, 2020, 页码6, 8, 13.

[20] Yanghao Li, Hanzi Mao, Ross Girshick, 和 Kaiming He. 探索用于目标检测的纯视觉变换器主干模型. arXiv预印本arXiv:2203.16527, 2022, 页码3, 7.

[21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, 和 Piotr Dollár. 使用卷积神经网络进行目标检测. 在国际计算机视觉会议(IEEE International Conference on Computer Vision), 2019, 第13条引用.

以上是我们根据相关文献内容进行的翻译，希望能对您有所帮助！

谢谢！

【22】Tsung-Yi Lin等人在2014年的欧洲计算机视觉会议上发表了名为《Microsoft coco: Common objects in context》的论文。该论文共有5页，起止页码为740-755。

【23】Shilong Liu等人在2022年的arXiv预印本中发表了名为《Dab-detr: Dynamic anchor boxes are better queries for detr》的论文。该论文的arXiv编号为arXiv:2201.12329。该论文涉及到了第2、3、6、7页。

【24】Ze Liu等人在2022年的IEEE/CVF计算机视觉与模式识别会议上发表了名为《Swin transformer v2: Scaling up capacity and resolution》的论文。该论文的页码范围为12009-12019，共有7页。

【25】Ze Liu等人在2021年的arXiv中发表了名为《Swin transformer: Hierarchical vision transformer using shifted windows》的论文。该论文的arXiv编号为abs/2103.14030。该论文涉及到了第2、6、7页。

【26】Depu Meng等人在2021年的IEEE/CVF国际计算机视觉会议上发表了名为《Conditional detr for fast training convergence》的论文。该论文的页码范围为3651-3660，共有6、7页。

【27】Shaoqing Ren等人在2015年的神经信息处理系统会议上发表了名为《Faster r-cnn: Towards real-time object detection with region proposal networks》的论文。该论文涉及到了第1、2、4、8、13页。

【28】Hamid Rezatofighi等人在2019年的arXiv预印本中发表了名为《Generalized intersection over union: A metric and a loss for bounding box regression》的论文。该论文的arXiv编号为arXiv:1902.09630。该论文涉及到了第7页。

box回归。在计算机视觉和模式识别的IEEE / CVF会议论文集上，页面658-666，2019年。

[29] Ramprasaath R Selvaraju，Michael Cogswell，Abhishek Das，Ramakrishna Vedantam，Devi Parikh和Dhruv Batra。Grad-cam：基于梯度的网络视觉解释。在计算机视觉的IEEE国际会议论文集上，页面618-626，2017年。

[30] 邵帅，李泽铭，张天远，彭超，于刚，张向宇，李敬和孙健。Objects365：用于物体检测的大规模高质量数据集。在计算机视觉的IEEE / CVF国际会议论文集上，页面8430-8439，2019年。

[31] 宋光陆，刘宇和王晓刚。重新审视目标检测器中的同级头。在计算机视觉和模式识别的IEEE / CVF会议论文集上，页面11563-11572，2020年。

[32] 田芷，沈春华，陈浩和何彤。FCOS：完全卷积单阶段目标检测。在计算机视觉的IEEE / CVF国际会议论文集上，页面9627-9636，2019年。

[33] 王文辉，包航波，董力，Johan Bjorck，彭志亮，刘强，Kriti Aggarwal，Owais Khan Mohammed，Saksham Singhal，Subhojit Som等。图像作为外语：Beit预训练适用于所有视觉和视觉语言任务。arXiv预印本arXiv:2208.10442，2022年。

[34] 王文海，戴季峰和陈喆。Internimage：2022年。

```
\documentclass{article}
\usepackage[UTF8]{ctex}
\begin{document}
\begin{thebibliography}{99}
  \bibitem{key36} Yixuan Wei, Han Hu, Zhenda Xie, Zheng Zhang, Yue Cao,
Jianmin Bao, Dong Chen, and Baining Guo. Contrastive
learning rivals masked image modeling in fine-tuning via
feature distillation. \textit{arXiv preprint arXiv:2205.14141}, 2022.
  \bibitem{key37} Zeyue Xue, Jianming Liang, Guanglu Song, Zhuofan Zong,
Liang Chen, Yu Liu, and Ping Luo. Large-batch optimization
for dense visual predictions. In \textit{Advances in Neural Information Processing Systems}, 2022.
  \bibitem{key38} Jianwei Yang, Chunyuan Li, and Jianfeng Gao. Focal modulation networks.
\textit{arXiv preprint arXiv:2203.11926}, 2022.
  \bibitem{key39} Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun
Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr
with improved denoising anchor boxes for end-to-end object
detection. \textit{arXiv preprint arXiv:2203.03605}, 2022.
  \bibitem{key40} Haotian Zhang, Pengchuan Zhang, Xiaowei Hu, Yen-Chun
Chen, Liunian Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-
Neng Hwang, and Jianfeng Gao. Glipv2: Unifying localiza-
tion and vision-language understanding. \textit{Advances in Neural Information Processing
Systems}, 35:36067–36080, 2022.
  \bibitem{key41} 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. \textit{Probabilistic two-stage
detection. arXiv preprint},
DETR ATSS
\end{thebibliography}
\end{document}
```

概率性的两阶段检测。arXiv预印本arXiv:2103.07461，2021年7月
[43] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. arXiv预印本arXiv:2010.04159，2020年2月6日，7日
[44] Zhuofan Zong, Qianggang Cao, and Biao Leng. Rcnet: Reverse feature pyramid and cross-

scale shift network for object detection. 第29届ACM国际多媒体会议论文集，第5637-5645页，2021年1月
11DETR的合作混合分配训练补充材料
＃对话0 1 2 3 4 5
AP 41.8 42.3 41.9 42.1 42.3 42.0
表12.辅助头部卷积数量的影响。
λ1 λ2＃训练周期AP APS APM APL
0.25 2.0 36 46.2 28.3 49.7 60.4
0.5 2.0 36 46.6 29.0 50.5 61.2
1.0 2.0 36 46.8 28.1 50.6 61.3
2.0 2.0 36 46.1 27.4 49.7 61.4
1.0 1.0 36 46.1 27.9 49.7 60.9
1.0 2.0 36 46.8 28.1 50.6 61.3
1.0 3.0 36 46.5 29.3 50.4 61.4
1.0 4.0 36 46.3 29.0 50.1 61.0
表13. λ1和λ2的超参数调整结果。
DETR ATSS
更快的RCNN DETR
ATSS
更快的RCNN 0.0000 0.0078 0.0085
0.0073 0.0000 0.0062
0.0079 0.0059 0.0000CAM KL散度
0.0000.0020.0040.0060.008
图8. DETR头部，ATSS头部和更快的RCNN头部的相关矩阵。检测器为Co-Deformable-DETR（K= 2）与ResNet-50。
A. 更多的消融研究
叠加卷积的数量。表12显示我们的方法对于叠加的卷积层数量具有鲁棒性
默认情况下将{λ1，λ2}设置为{1.0,2.0}。
ATSS

达到更高性能的能力轻量级。
合作训练的损失权重。与权重系数λ1和λ2相关的实验结果
显示在表13中。我们发现，由于性能在变化损失系数时稍微波动，
所以这种提议的方法对{λ1，λ2}的变化具有较高的鲁棒性。
总之，系数{λ1，λ2}稳健可靠，我们将{λ1，λ2}默认设置为
{1.0,2.0}。
DETR ATSS
Faster-RCNN PAA GFL FCOS
RetinaNet
DETR
ATSS
Faster-RCNN
PAA
GFL
FCOS
RetinaNet
0.0000 0.0062 0.0090 0.0129 0.0079 0.0069 0.0083
0.0066 0.0000 0.0097 0.0132 0.0077 0.0045 0.0083
0.0083 0.0087 0.0000 0.0091 0.0055 0.0095 0.0063
0.0116 0.0116 0.0083 0.0000 0.0075 0.0125 0.0090
0.0073 0.0068 0.0056 0.0083 0.0000 0.0081 0.0054
0.0075 0.0048 0.0107 0.0143 0.0092 0.0000 0.0092

0.0078 0.0075 0.0066 0.0100 0.0055 0.0082 0.0000
CAM KL散度
0.0000.0020.0040.0060.0080.0100.0120.014图9.我们模型中7个不同头的距离
（K= 6）。
自定义正样本查询数量。我们计算一个对多标签
分配中正样本的平均比例，与真实框相比。例如，Faster-RCNN
的比例为18.7，ATSS的比例为8.8，这意味着在K= 1时引入了超过8倍的额外正样本查询。
协作式一对多标签分配的有效性。为验证我们的特征学习机制
的有效性，我们将我们的方法与Group-DETR学习进行比较。
Si,j=1
|D|X
Si=1
13

没有定制的正样本查询生成。更重要的是，图2中的IoF-IoB曲线证明Group-DETR未能增强编码器中的特征表示，而我们的方法缓解了特征学习不良的问题。
冲突分析。我们在这项研究中定义了头部Hi和头部Hj之间的距离，并使用Hito的平均距离来衡量优化冲突：
Si,j=1
|D|X
I∈DKL(C(Hi(I)),C(Hj(I)), (11)
Si=1
2(K−1)KX
j=i(Si,j+Sj,i), (12)
其中KL,D,I,C分别指代KL散度、数据集、输入图像和类别激活图(CAM) [29]。在我们的实现中，我们选择验证集COCO val作为D，选择Grad-CAM作为C。我们使用DETR编码器的输出特征来计算CAM图。具体而言，我们在图8和图9中展示了K=2和K=6时的详细距离。Si,j的较大距离指示Hi与Hj less一致，并对优化不一致性做出贡献。
B.更多实现细节
一阶辅助头。基于常规的一阶检测器，我们尝试了各种第一阶段的设计[17,19,21,32,41]作为辅助头。首先，我们使用GIoU [28]损失函数用于一阶头。然后，将堆叠的卷积数量从4个减少到1个。这样的修改在不降低准确性的情况下提高了训练效率。对于无锚点检测器，例如FCOS [32]，我们使用步长2j为正坐标分配宽度为8×2j和高度为8×2j。

两阶段辅助头部。我们采用了RPN和RCNN作为我们的两阶段辅助头部，基于流行的Faster-RCNN [27]和Mask-RCNN [14]检测器。为了使Co-DETR与各种检测头兼容，我们采用了与一阶段范式相同的多尺度特征（步幅从8到128）作为两阶段辅助头部。此外，我们在RCNN阶段采用了GIoU回归损失。

COCO的系统级比较。我们首先使用EV A-02权重初始化ViT-L主干网络。然后，我们使用Co-DINO-Deformable-DETR在Objects365数据集上进行26个epoch的中间微调，并在第24个epoch时将学习率降低了0.1倍。初始学习率为2.5×10−4，批大小为224。我们将输入图像的最大尺寸设为1280，并随机调整较短边的尺寸在480至1024之间。此外，我们为该模型使用了1500个物体查询和1000个DN查询。最后，我们在COCO上对Co-DETR进行12个epoch的微调，初始学习率为5×10−5，并在第8个epoch时通过乘以0.1降低学习率。输入图像的较短边被调整为480至1536，而较长边不超过2400。我们使用EMA进行训练，并使用批大小为64。

LVIS的系统级比较。与COCO设置相反，我们使用Co-DINO-Deformable-DETR++在Objects365数据集上进行中间微调，因为我们发现LSJ数据增强在LVIS数据集上效果更好。批大小为192，初始学习率为13。

我们对该模型使用了900个对象查询和1000个DN查询。在对LVIS进行微调时，我们为其配备了一个额外的辅助掩模分支，并增加了输入大小至1536×1536。此外，我们在不使用EMA的情况下对模型进行了16个时期的训练，批量大小设置为64，初始学习率设置为5×10−5，这在第9和第15个时期时减少了0.1倍。