Rethinking Transformer-based Set Prediction for Object Detection

Zhiqing Sun* Shengcao Cao* Yiming Yang Kris Kitani
Carnegie Mellon University
{zhiqings,shengcao,yiming,kkitani}@cs.cmu.edu

Abstract

DETR is a recently proposed Transformer-based method which views object detection as a set prediction problem and achieves state-of-the-art performance but demands extra-long training time to converge. In this paper, we investigate the causes of the optimization difficulty in the training of DETR. Our examinations reveal several factors contributing to the slow convergence of DETR, primarily the issues with the Hungarian loss and the Transformer cross-attention mechanism. To overcome these issues we propose two solutions, namely, TSP-FCOS (Transformer-based Set Prediction with FCOS) and TSP-RCNN (Transformer-based Set Prediction with RCNN). Experimental results show that the proposed methods not only converge much faster than the original DETR, but also significantly outperform DETR and other baselines in terms of detection accuracy. Code is released at `https://github.com/Edward-Sun/TSP-Detection`.

1. Introduction

Object detection aims at finding all objects of interest in an image and predicting their category labels and bounding boxes, which is essentially a set prediction problem, as the ordering of the predicted objects is not required. Most of the state-of-the-art neural detectors [21, 24, 19, 25, 38, 26, 11] are developed in a detect-and-merge fashion that is, instead of directly optimizing the predicted set in an end-to-end fashion, those methods usually first make predictions on a set of region proposals or sliding windows, and then perform a post-processing step (e.g., "non-maximum suppression" or NMS) for merging the the detection results in different proposals or windows that might belong to the same object. As the detection model is trained agnostically with respect to the merging step, the model optimization in those object detectors is not end-to-end and arguably sub-optimal.

DEtection TRansformer (DETR) [2] is recently proposed as the first fully end-to-end object detector. It uses Transformer [32] to directly output a final set of predictions without further post-processing. However, it takes extra-long training time to converge. For example, the popular Faster RCNN model [26] only requires about 30 epochs to convergence, but DETR needs 500 epochs, which takes at least 10 days on 8 V100 GPUs. Such expensive training cost would be practically prohibitive in large applications. Therefore, in what manner should we accelerate the training process towards fast convergence for DETR-like Transformer-based detectors is a challenging research question and is the main focus of this paper.

For analyzing the causes of DETR's optimization difficulty we conduct extensive experiments and find that the cross-attention module, by which the Transformer decoder obtains object information from images, is mainly responsible for the slow convergence. In pursuit of faster convergence, we further examine an encoder-only version of DETR by removing the cross-attention module. We find that the encoder-only DETR yields a substantial improvement for the detection of small objects in particular but sub-optimal performance on large objects. In addition, our analysis shows that the instability of the bipartite matching in DETR's Hungarian loss also contributes to the slow convergence.

Based on the above analysis we propose two models for significantly accelerating the training process of Transformer-based set prediction methods, both of which can be regarded as improved versions of encoder-only DETR with feature pyramids [18]. Specifically, we present TSP-FCOS (Transformer-based Set Prediction with FCOS) and TSP-RCNN (Transformer-based Set Prediction with RCNN), which are inspired by a classic one-stage detector FCOS [30] (Fully Convolutional One-Stage object detector) and a classic two-stage detector Faster RCNN [26], respectively. A novel Feature of Interest (FoI) selection mechanism is developed in TSP-FCOS to help Transformer encoder handle multi-level features. To resolve the instability of the bipartite matching in the Hungarian loss, we also design a new bipartite matching scheme for each of our two models for accelerating the convergence in training. In our evaluation on the COCO 2017 detection benchmark [20]

*indicates equal contribution.

重新思考基于Transformer的集合预测在目标检测中的应用

孙志清* 曹胜操* 杨一鸣 克里斯·北谷 卡内基梅隆大学

{zhiqings,shengcao,yiming,kkitani}@cs.cmu.edu

摘要

DETR is a recently proposed Transformer-based method which views object detection as a set prediction problem and achieves state-of-the-art performance but demands extra-long training time to converge. In this paper, we investigate the causes of the optimization difficulty in the training of DETR. Our examinations reveal several factors contributing to the slow convergence of DETR, primarily the issues with the Hungarian loss and the Transformer cross-attention mechanism. To overcome these issues we propose two solutions, namely, TSP-FCOS (Transformer-based Set Prediction with FCOS) and TSP-RCNN (Transformer-based Set Prediction with RCNN). Experimental results show that the proposed methods not only converge much faster than the original DETR, but also significantly outperform DETR and other baselines in terms of detection accuracy. Code is released at https://github.com/ Edward-Sun/TSP-Detection.

1. 引言

目标检测旨在找出图像中所有感兴趣的目标，并预测其类别标签和边界框，这本质上是一个集合预测问题，因为预测目标的顺序并不重要。大多数先进的神经检测器[21, 24, 19, 25, 38, 26, 11]采用"检测-合并"的开发模式——这些方法并非直接以端到端方式优化预测集合，而是先对一组候选区域或滑动窗口进行预测，然后通过后处理步骤（如"非极大值抑制"NMS）合并可能属于同一目标的不同候选框或窗口的检测结果。由于检测模型的训练与合并步骤无关，这类目标检测器的模型优化并非端到端，因而可能存在次优问题。

DEtection TRansformer (DETR) [2] 是近期提出的首个完全端到端目标检测器。它采用Transformer [32]能够直接输出最终的预测集合，无需进一步后处理。然而，其收敛所需的训练时间异常漫长。例如，广为人知的Faster RCNN模型[26]仅需约30个训练周期即可收敛，而DETR则需要500个周期，即便使用8块V100 GPU也至少耗时10天。如此高昂的训练成本在大型实际应用中往往难以承受。因此，如何加速此类基于Transformer的DETR检测器的训练过程以实现快速收敛，便成为一个极具挑战性的研究课题，也是本文的核心关注点。

为分析DETR优化困难的成因，我们进行了大量实验，发现Transformer解码器通过交叉注意力模块从图像中获取目标信息，该模块是导致收敛缓慢的主要原因。为追求更快收敛，我们进一步研究了移除交叉注意力模块后的纯编码器版DETR。研究发现，纯编码器DETR对小目标检测效果提升显著，但对大目标检测性能欠佳。此外，分析表明DETR匈牙利损失中二分匹配的不稳定性也是造成收敛缓慢的原因之一。

基于上述分析，我们提出了两种显著加速基于Transformer的集合预测方法训练过程的模型，二者均可视为带有特征金字塔的纯编码器DETR[18]的改进版本。具体而言，我们提出了受经典一阶段检测器FCOS[30]（全卷积一阶段目标检测器）启发的TSP-FCOS（基于Transformer的FCOS集合预测）和受经典两阶段检测器Faster RCNN[26]启发的TSP-RCNN（基于Transformer的RCNN集合预测）。TSP-FCOS中开发了新颖的感兴趣特征（FoI）选择机制，以帮助Transformer编码器处理多层级特征。针对匈牙利损失中二分图匹配的不稳定性问题，我们还为两个模型分别设计了新的二分图匹配方案以加速训练收敛。在COCO 2017检测基准[20]上的评估中

*indicates equal contribution.

the proposed methods not only converge much faster than the original DETR, but also significantly outperform DETR and other baselines in terms of detection accuracy.

2. Background

2.1. One-stage and Two-stage Object Detectors

Most modern object detection methods can be divided into two categories: One-stage detectors and two-stage detectors. Typical one-stage detectors [21, 24, 19] directly make predictions based on the extracted feature maps and (variable-sized) sliding-window locations in a image, while two-stage detectors [26, 11] first generate region proposals based on sliding-window locations and then refine the detection for each proposed region afterwards. In general, two-stage detectors are more accurate but also computationally more expensive than one-stage detectors. Nevertheless, both kinds of detectors are developed in a detection-and-merge fashion, *i.e*., they require a post-processing step to ensure that each detected object has only one region instead of multiple overlapping regions as detection results. In other words, many state-of-the-art object detection methods do not have an end-to-end training objective with respect to set prediction.

2.2. DETR with an End-to-end Objective

Different from the aforementioned popular object detectors, DEtection TRansformer (DETR) [2] presents the first method with an end-to-end optimization objective for set prediction. Specifically, it formulates the loss function via a bipartite matching mechanism. Let us denote by $y = \{y_i\}_{u=1}^{M}$ the ground truth set of objects, and $\hat{y} = \{\hat{y}_i\}_{u=1}^{N}$ the set of predictions. Generally we have $M < N$, so we pad y to size N with $\emptyset$ (no object) and denote it by $\bar{y}$. The loss function, namely the Hungarian loss, is defined as:

$$\mathcal{L}_{\text{Hungarian}}(\bar{y}, \hat{y}) = \sum_{i=1}^{N} \left[\mathcal{L}_{class}^{i,\hat{\sigma}(i)} + \mathbb{1}_{\{\bar{y}_i \neq \emptyset\}} \mathcal{L}_{box}^{i,\hat{\sigma}(i)} \right] \quad (1)$$

where $\mathcal{L}_{class}^{i,\hat{\sigma}(i)}$ and $\mathcal{L}_{box}^{i,\hat{\sigma}(i)}$ are the classification loss and bounding box regression loss, respectively, between the i^{th} ground truth and the $\hat{\sigma}(i)^{th}$ prediction. And $\hat{\sigma}$ is the optimal bipartite matching between padded ground-truth set $\bar{y}$ and prediction set $\hat{y}$ with lowest matching cost:

$$\hat{\sigma} = \arg\min_{\sigma \in \mathfrak{S}_N} \sum_{i=1}^{N} \mathcal{L}_{\text{match}}(\bar{y}_i, \hat{y}_{\sigma(i)}) \quad (2)$$

where $\mathfrak{S}_N$ is the set of all N permutations and $\mathcal{L}_{\text{match}}$ is a pair-wise matching cost.

DETR [2] uses an encoder-decoder Transformer [32] framework built upon the CNN backbone. The Transformer encoder part processes the flattened deep features[1] from the CNN backbone. Then, the non-autoregressive decoder part takes the encoder's outputs and a set of learned object query vectors as the input, and predicts the category labels and bounding boxes accordingly as the detection output. The cross-attention module plays an important role in the decoder by attending to different locations in the image for different object queries. We refer the reader unfamiliar with the Transformer concepts to the appendix. The attention mechanism in DETR eliminates the need for NMS post-processing because the self-attention component can learn to remove duplicated detection, *i.e*., its Hungarian loss (Equation 1) encourages one target per object in the bipartite matching.

Concurrent to our work, some variants of DETR have been proposed to improve its training efficiency and accuracy. Deformable DETR [42] proposes to integrate the concept of deformable convolution and attention modules, to implement a sparse attention mechanism on multi-level feature maps. UP-DETR [8] leverages an unsupervised pre-training task named random query patch detection to improve the performance of DETR when it is fine-tuned on down-stream tasks. Compared to these work, we explore further simplifying the detection head design with encoder-only Transformer.

2.3. Improving Ground-truth Assignments

The Hungarian loss in DETR can be viewed as an end-to-end way to assign ground-truth labels to the system predictions. Prior to DETR, heuristic rules have been tried for this task [10, 26, 24]. There are a few other prior work that try to improve the heuristic ground-truth assignment rules. [39] formulates an MLE procedure to learn the matching between sliding windows and ground truth objects. [27] proposes a generalized IoU which provides a better metric. Nevertheless, those methods do not directly optimize a set-based objective and still require an NMS post-processing step.

2.4. Attention-based Object Detection

Attention-based modeling has been the current workhorse in the Natural Language Processing (NLP) domain [32, 9], and is becoming increasingly popular in recent object detection research. Before the invention of DETR, [12] proposes an attention-based module to model the relation between objects, which can be inserted into existing detectors and leads to better recognition and less duplication. [23] uses a Spatial Attention Module to re-weight feature maps for making foreground features standing out. [4] uses a Transformer-like attention-based module to bridge different forms of representations.

[1] In this paper, we use "feature points" and "features" interchangeably.

所提出的方法不仅收敛速度远超原始DETR，而且在检测精度方面也显著优于DETR及其他基线模型。

2. 背景

2.1. 单阶段与双阶段目标检测器

现代大多数目标检测方法可分为两大类：单阶段检测器与两阶段检测器。典型的单阶段检测器[21,24,19]直接基于提取的特征图及图像中（可变尺寸的）滑动窗口位置进行预测，而两阶段检测器[26,11]首先生成基于滑动窗口位置的区域提议，随后对每个提议区域进行检测优化。总体而言，两阶段检测器精度更高，但计算成本也显著高于单阶段检测器。然而，这两类检测器均采用检测后合并的开发范式*i.e*，即需要通过后处理步骤确保每个检测目标仅对应一个区域，而非作为检测结果的多个重叠区域。换言之，许多前沿目标检测方法并未建立针对集合预测的端到端训练目标。

2.2. 采用端到端目标的DETR

与上述流行的目标检测器不同，DEtection TRansformer (DETR) [2] 首次提出了一种端到端优化目标的方法，用于集合预测。具体而言，它通过二分图匹配机制构建损失函数。设真实目标集合为$y = \{y_i\}_{u=1}^{M}$，预测集合为$\hat{y} = \{\hat{y}_i\}_{u=1}^{N}$。通常我们拥有$M < N$，因此我们将$y$填充至$N$大小，并用$\emptyset$ (无目标)表示，记为$\bar{y}$。该损失函数，即匈牙利损失，定义为：

$$\mathcal{L}_{\text{Hungarian}}(\bar{y}, \hat{y}) = \sum_{i=1}^{N} \left[\mathcal{L}_{class}^{i,\hat{\sigma}(i)} + \mathbb{1}_{\{\bar{y}_i \neq \emptyset\}} \mathcal{L}_{box}^{i,\hat{\sigma}(i)} \right] \quad (1)$$

其中$\mathcal{L}_{class}^{i,\hat{\sigma}(i)}$和$\mathcal{L}_{box}^{i,\hat{\sigma}(i)}$分别表示$i^{th}$真实标注与$\hat{\sigma}(i)^{th}$预测结果之间的分类损失和边界框回归损失。而$\hat{\sigma}$则是填充后的真实标注集$\bar{y}$与预测集$\hat{y}$之间具有最低匹配成本的最优二分匹配：

$$\hat{\sigma} = \underset{\sigma \in \mathfrak{S}_N}{\arg\min} \sum_{i=1}^{N} \mathcal{L}_{\text{match}}(\bar{y}_i, \hat{y}_{\sigma(i)}) \quad (2)$$

其中$\mathfrak{S}_N$是所有N排列的集合，$\mathcal{L}_{\text{match}}$是两两匹配成本。

DETR [2]采用了基于CNN骨干网络的编码器-解码器Transformer [32]框架。该Transformer编码器部分处理来自CNN主干的扁平化深度特征[1]。随后，非自回归解码器部分以编码器的输出和一组学习得到的目标查询向量作为输入，预测类别标签和边界框作为检测输出。解码器中的交叉注意力模块通过为不同目标查询关注图像中的不同位置发挥关键作用。对Transformer概念不熟悉的读者可参阅附录。DETR中的注意力机制消除了对NMS后处理的需求，因为自注意力组件能够学会去除重复检测*i.e*，其匈牙利损失（公式1）在二分图匹配中促使每个目标对应一个预测结果。

与我们的工作同时，一些DETR的变体被提出以提升其训练效率和准确性。可变形DETR[42]提出整合可变形卷积与注意力模块的概念，在多层次特征图上实现稀疏注意力机制。UP-DETR[8]则利用一种名为随机查询块检测的无监督预训练任务，来改善DETR在下游任务微调时的表现。相较于这些工作，我们探索了仅使用编码器的Transformer进一步简化检测头设计。

2.3. 优化真实值分配

DETR中的匈牙利损失可视为一种端到端的方式，用于为系统预测分配真实标签。在DETR之前，已有研究尝试采用启发式规则完成此任务[10,26,24]。另有若干早期工作致力于改进启发式真值分配规则：[39]提出最大似然估计程序来学习滑动窗口与真实物体间的匹配关系；[27]提出广义交并比指标以提供更优度量标准。然而这些方法均未直接优化基于集合的目标函数，且仍需依赖非极大值抑制后处理步骤。

2.4. 基于注意力的目标检测

基于注意力的建模已成为自然语言处理（NLP）领域当前的主流方法[32, 9]，并在近年来的目标检测研究中日益流行。在DETR问世之前，[12]提出了一种基于注意力的模块来建模物体间关系，该模块可嵌入现有检测器中，从而实现更好的识别效果并减少重复检测。[23]采用空间注意力模块对特征图进行重加权，使前景特征更为突出。[4]则利用类Transformer的注意力模块来桥接不同形式的表征。

[1] In this paper, we use “feature points” and “features” interchangeably.

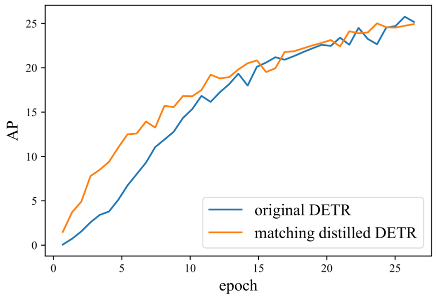

Figure 1. AP results on COCO validation set: original DETR v.s. matching distilled DETR. We can see that matching distillation accelerates the training of DETR in the first few epochs.

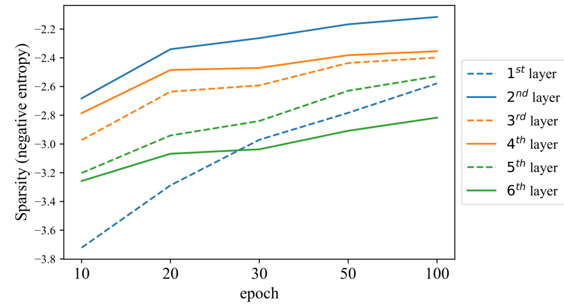

Figure 2. Sparsity (negative entropy) of Transformer cross-attention in each layer, obtained by evaluation on the COCO validation data. Different line styles represent different layers. We can see that the sparsity consistently increases, especially for the 1^{st} cross-attention layer between encoder and decoder.

But, none of those methods have tried an end-to-end set prediction objective.

3. What Causes the Slow Convergence of DETR?

To pin down the main factors we ran a set of experiments with DETR and its variants which are built on top of the ResNet-50 backbone and evaluated on the COCO 2017 validation set.

3.1. Does Instability of the Bipartite Matching Affect Convergence?

As a unique component in DETR, the Hungarian loss based on the bipartite matching (Section 2.2) could be unstable due to the following reasons:

- The initialization of the bipartite matching is essentially random;
- The matching instability would be caused by noisy conditions in different training epochs.

To examine the effects of these factors, we propose a new training strategy for DETR, namely matching distillation. That is, we use a well pre-trained DETR as the teacher model, whose predicted bipartite matching is treated as the ground-truth label assignment for the student model. All stochastic modules in the teacher model (*i.e.*, dropout [29] and batch normalization [13]) are turned off to ensure the provided matching is deterministic, which eliminates the randomness and instability of the bipartite matching and hence in the Hungarian loss.

We evaluated both the original DETR and matching distilled DETR. Figure 1 shows the results with the first 25 epochs. We can see that the matching distillation strategy does help the convergence of DETR in the first few epochs. However, such effect becomes insignificant after around 15 epochs. This means that the instability in the bipartite matching component of DETR only contributes partially to the slow convergence (especially in the early training stage) but not necessarily the main reason.

3.2. Are the Attention Modules the Main Cause?

Another distinct part of DETR in comparison with other modern object detectors is its use of the Transformer modules, where the Transformer attention maps are nearly uniform in the initialization stage, but gradually become more and more sparse during the training process towards the convergence. Prior work [14] shows that replacing some attention heads in BERT [9] with sparser modules (*e.g.*, convolutions) can significantly accelerate its training. Therefore, it is natural for us to wonder how much the sparsity dynamics of Transformer attention modules in DETR contribute to its slow convergence.

In analyzing the effects of the DETR's attention modules on its optimization convergence, we focus on the sparsity dynamics of the cross-attention part in particular, because the cross-attention module is a crucial module where object queries in the decoder obtain object information from the encoder. Imprecise (under-optimized) cross-attention may not allow the decoder to extract accurate context information from images, and thus results in poor localization especially for small objects.

We collect the attention maps of cross-attention when evaluating the DETR model at different training stages. As attention maps can be interpreted as probability distributions, we use negative entropy as an intuitive measure of sparsity. Specifically, given a $n \times m$ attention map $\mathbf{a}$, we first calculate the sparsity of each source position $i \in [n]$ by $\frac{1}{m}\sum_{j=1}^{m} P(a_{i,j}) \log P(a_{i,j})$, where $a_{i,j}$ represents the attention score from source position i to target position j. Then we average the sparsities for all attention heads and all source positions in each layer. The masked positions [2] are not considered in the computation of sparsity.

Figure 2 shows the sparsities with respect to different epochs at several layers. we can see that the sparsity of cross-attention consistently increases and does not reach a plateau even after 100 training epochs. This means that

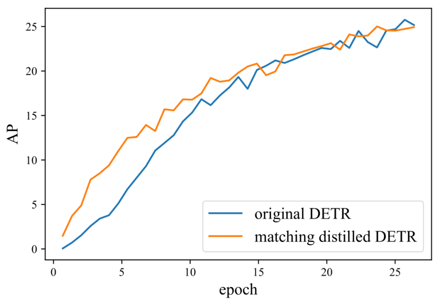

图1. COCO验证集上的AP结果：原始DETR对比匹配蒸馏DETR。可以看出，匹配蒸馏在最初几个训练周期内加速了DETR的训练。

但是，这些方法中没有一个尝试过端到端的集合预测目标。

3. DETR收敛缓慢的原因是什么?

为了确定主要因素，我们使用基于ResNet-50骨干网络构建的DETR及其变体进行了一系列实验，并在COCO 2017验证集上进行了评估。

3.1. 二分匹配的不稳定性是否会影响收敛?

作为DETR中一个独特的组成部分，基于二分图匹配的匈牙利损失（第2.2节）可能因以下原因而不稳定:

- 二分匹配的初始化本质上是随机的;
- 匹配不稳定性可能由不同训练周期中的噪声条件引起。

为了探究这些因素的影响，我们为DETR提出了一种新的训练策略——匹配蒸馏。具体而言，我们采用一个预训练良好的DETR作为教师模型，其预测的双边匹配结果将作为学生模型的真实标签分配。教师模型中所有随机性模块（*i.e*，如dropout[29]和批量归一化[13]）均被关闭，以确保提供的匹配具有确定性，从而消除双边匹配及匈牙利损失中的随机性与不稳定性。

我们评估了原始DETR与匹配蒸馏后的DETR。图1展示了前25个epoch的结果。可以看出，匹配蒸馏策略确实在最初几个epoch中促进了DETR的收敛。然而，这种效果在大约15个epoch后变得不再显著。这表明DETR中二分匹配组件的不稳定性仅部分导致了收敛缓慢（尤其在训练初期），而不一定是主要原因。

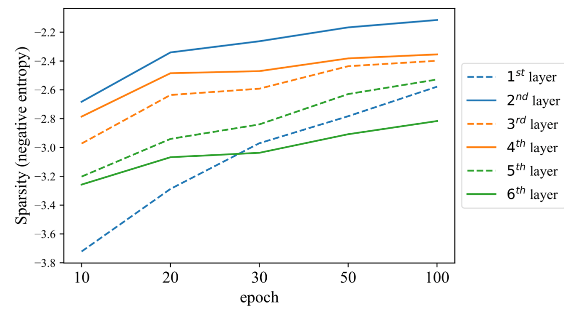

图2. Transformer交叉注意力在各层的稀疏性（负熵），通过在COCO验证数据上的评估得出。不同线型代表不同层。可以看出，稀疏性持续增加，尤其是编码器与解码器之间的1^{st}交叉注意力层。

3.2. 注意力模块是主要原因吗?

与其他现代目标检测器相比，DETR另一个显著特点是其采用了Transformer模块。在初始化阶段，Transformer的注意力图几乎呈均匀分布，但随着训练过程向收敛推进，这些注意力图会逐渐变得越来越稀疏。先前的研究[14]表明，在BERT[9]中用更稀疏的模块（{v*}，如卷积）替换部分注意力头，能显著加速训练。因此，我们很自然地会思考：DETR中Transformer注意力模块的稀疏性动态变化，对其缓慢收敛的贡献程度究竟有多大。

在分析DETR注意力模块对其优化收敛性的影响时，我们特别关注交叉注意力部分的稀疏性动态，因为交叉注意力模块是解码器中对象查询从编码器获取对象信息的关键组件。不精确（未充分优化）的交叉注意力可能导致解码器无法从图像中提取准确的上下文信息，从而造成定位效果不佳，尤其是对小物体而言。

我们在评估DETR模型不同训练阶段时，收集了交叉注意力机制的注意力图。由于注意力图可被解读为概率分布，我们采用负熵作为稀疏性的直观度量指标。具体而言，给定一个$n \times m$注意力图$\mathbf{a}$，首先通过$\frac{1}{m}\sum_{j=1}^{m} P(a_{i,j}) \log P(a_{i,j})$计算每个源位置$i \in [n]$的稀疏度，其中$a_{i,j}$表示从源位置$i$到目标位置$j$的注意力分数。随后我们对每层中所有注意力头及所有源位置的稀疏度取平均值。计算稀疏度时，掩码位置[2]不予考虑。

图2展示了不同训练周期下多个层的稀疏性情况。可以看出，交叉注意力机制的稀疏性持续上升，即便在训练超过100个周期后仍未达到稳定状态。这意味着

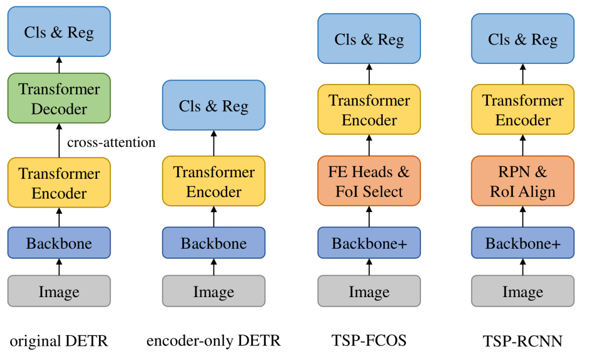

Figure 3. Illustration of original DETR, encoder-only DETR, TSP-FCOS, and TSP-RCNN, where Backbone+, FE Heads, RPN, Cls & Reg represents "Backbone + FPN", "Feature Extraction Heads (Subnets)", "Region Proposal Network", "Classification & Regression", respectively. A more detailed illustration of TSP-FCOS and TSP-RCNN can be found in Figure 5.

the cross-attention part of DETR is more dominating a factor for the slow convergence, compared to the early-stage bipartite-matching instability factor we discussed before.

3.3. Does DETR Really Need Cross-attention?

Our next question is: Can we remove the cross-attention module from DETR for faster convergence but without sacrificing its prediction power in object detection? We answer this question by designing an encoder-only version of DETR and comparing its convergence curves with the original DETR.

In the original DETR, the decoder is responsible for producing the detection results (category label and bounding box) per object query. In contrast, the encoder-only version of DETR (introduced by us) directly uses the outputs of Transformer encoder for object prediction. Specifically, for a $H \times W$ image with a $\frac{H}{32} \times \frac{W}{32}$ Transformer encoder feature map, each feature is fed into a detection head to predict a detection result. Since the encoder self-attention is essentially identical to the self-attention in a non-autoregressive decoder, a set prediction training is still feasible for encoder-only DETR. More details of encoder-only DETR can be found in the appendix. Figure 3 compares the original DETR and the encoder-only DETR, and two of our newly proposed models (TSP-FCOS and TSP-RCNN) which are described in the next section.

Figure 4 presents the the Average Precision (AP) curves of the original DETR and the encoder-only DETR, including the overall AP curve (denoted as AP) and the curves for large (AP-l), medium (AP-m), and small (AP-s) objects[2], respectively. The over-all curves (left upper corner) show that the encoder-only DETR performs as well as the original DETR. This means that we can remove the cross-attention part from DETR without much performance degeneration, which is a positive result. From the remaining curves we can see that the encoder-only DETR outperforms the original DETR significantly on small objects and partly on medium object, but under-performs on large objects on the other hand. A potential interpretation, we think, is that a large object may include too many potentially matchable feature points, which are difficult for the sliding point scheme in the encoder-only DETR to handle. Another possible reason is that a single feature map processed by encoder is not robust for predicting objects of different scales [18].

[2] We follow the definitions of small, medium, and large objects in [20].

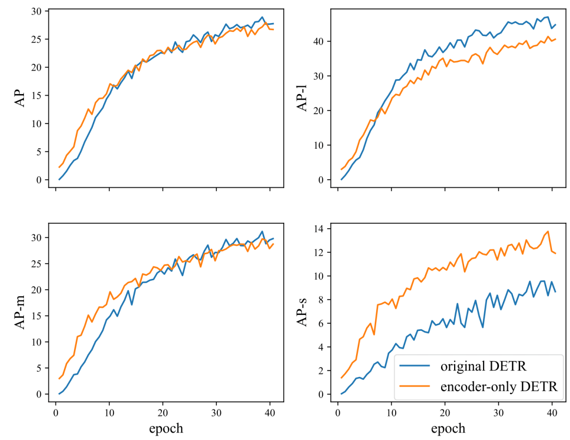

Figure 4. AP, AP-l, AP-m, and AP-s results on COCO validation set: original DETR v.s. encoder-only DETR. We can see that encoder-only DETR significantly accelerate the training of small object detection.

4. The Proposed Methods

Based on our analysis in the previous section, for speeding up the convergence of DETR we need to address both the instability issue in the bipartite matching part of DETR and the cross-attention issue in Transformer modules. Specifically, in order to leverage the speed-up potential of encoder-only DETR we need to overcome its weakness in handling the various scales of objects. Recently, FCOS [30] (Fully Convolutional One-Stage object detector) shows that multi-level prediction with Feature Pyramid Network (FPN) [18] is a good solution to this problem. Inspired by this work we propose our first model, namely Transformer-based Set Prediction with FCOS (TSP-FCOS). Then based on TSP-FCOS, we further apply two-stage refinement, which leads to our second model, namely, Transformer-based Set Prediction with RCNN (TSP-RCNN).

4.1. TSP-FCOS

TSP-FCOS combines the strengths of both FCOS and encoder-only DETR, with a novel component namely Fea-

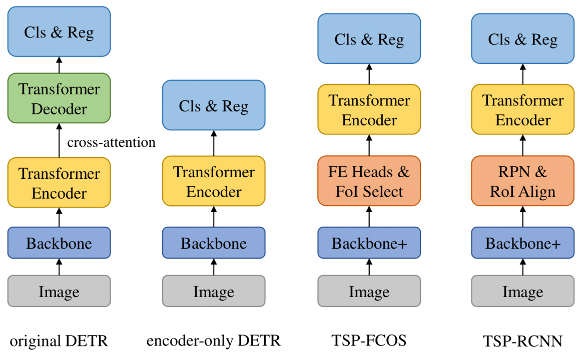

图3. 原始DETR、仅编码器DETR、TSP-FCOS和TSP-RCNN的示意图，其中Backbone+、FE Heads、RPN、Cls & Reg分别代表“Backbone + FPN”、“特征提取头（子网络）”、“区域提议网络”、“分类与回归”。TSP-FCOS和TSP-RCNN的更详细图示可参见图5。

与之前讨论的早期二分匹配不稳定性因素相比，DETR中的交叉注意力部分对收敛速度缓慢的影响更为显著。

3.3. DETR真的需要交叉注意力吗?

我们的下一个问题是：能否从DETR中移除交叉注意力模块以加速收敛，同时不牺牲其在目标检测中的预测能力？我们通过设计一个仅含编码器版本的DETR，并将其收敛曲线与原版DETR进行对比，来回答这一问题。

在原始的DETR中，解码器负责为每个对象查询生成检测结果（类别标签和边界框）。相比之下，我们提出的仅编码器版本DETR直接利用Transformer编码器的输出进行对象预测。具体而言，对于具有$\frac{H}{32} \times \frac{W}{32}$ Transformer编码器特征图的$H \times W$图像，每个特征会被送入检测头以预测检测结果。由于编码器自注意力机制本质上与非自回归解码器中的自注意力相同，因此集合预测训练对仅编码器DETR仍然可行。更多关于仅编码器DETR的细节可在附录中找到。图3对比了原始DETR、仅编码器DETR以及我们新提出的两种模型（TSP-FCOS和TSP-RCNN），后两者将在下一节详述。

图4展示了原始DETR与仅编码器DETR的平均精度（AP）曲线，包括整体AP曲线（标记为AP）以及分别针对大（AP-l）、中（AP-m）、小（AP-s）物体[2]的曲线。总体曲线（左上角）显示，仅编码器DETR的表现与原始DETR相当。这意味着我们可以从DETR中移除交叉注意力部分而不会显著影响性能——

[2]We follow the definitions of small, medium, and large objects in [20].

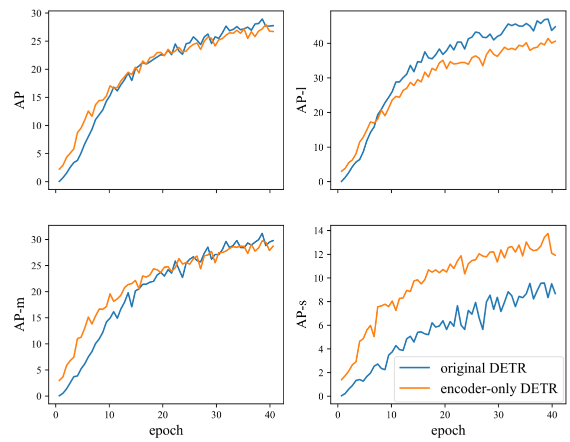

图4. COCO验证集上的AP、AP-l、AP-m和AP-s结果：原始DETR对比仅编码器DETR。可以看出，仅编码器DETR显著加快了小物体检测的训练速度。

生成，这是一个积极的结果。从剩余的曲线可以看出，仅编码器的DETR在小物体上显著优于原始DETR，在中型物体上也有部分优势，但在大物体上表现则相对欠佳。我们认为一个可能的解释是，大物体可能包含过多潜在可匹配的特征点，这对仅编码器DETR中的滑动点方案来说难以处理。另一个可能的原因是，编码器处理的单一特征图对于预测不同尺度的物体不够鲁棒[18]。

4. 所提出的方法

根据我们在前一节的分析，为了加速DETR的收敛，需要同时解决DETR中二分图匹配部分的不稳定性问题以及Transformer模块中的交叉注意力问题。具体而言，为了充分发挥仅含编码器的DETR在加速上的潜力，必须克服其在处理不同尺度物体时的不足。近期，FCOS[30]（全卷积单阶段目标检测器）表明，结合特征金字塔网络（FPN）[18]的多层级预测是解决这一问题的有效方案。受此启发，我们提出了首个模型——基于Transformer的FCOS集合预测（TSP-FCOS）。随后，在TSP-FCOS基础上进一步应用两阶段优化策略，由此衍生出第二个模型——基于Transformer的RCNN集合预测（TSP-RCNN）。

4.1. TSP-FCOS

TSP-FCOS融合了FCOS与仅编码器DETR的优势，引入了一个创新组件——Fea-

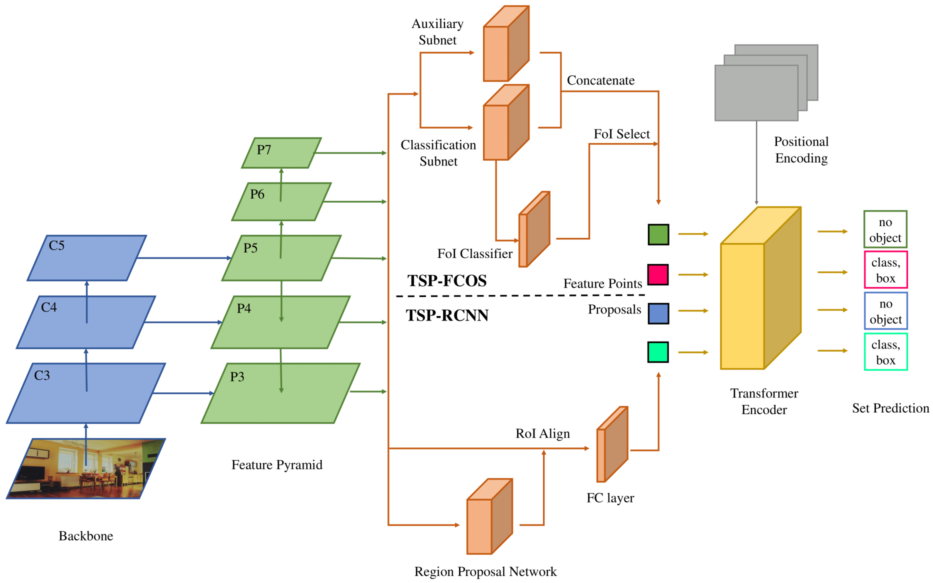

Figure 5. The network architectures of TSP-FCOS and TSP-RCNN, where C_3 to C_5 denote the feature maps of the backbone network and P_3 to P_7 of the Feature Pyramid Network (FPN). Both TSP-FCOS (upper) and TSP-RCNN (lower) are equipped with a Transformer encoder and trained with set prediction loss. The difference between them is that the FoI classifier in TSP-FCOS only predicts objectness of each feature (*i.e.*, Features of Interest), but the Region Proposal Network (RPN) in TSP-RCNN predicts both bounding boxes and their objectness as Regions of Interest (RoI), namely, proposals.

ture of Interest (FoI) selection which enables the Transformer encoder to handle multi-level features, and a new bipartite matching scheme for faster set prediction training. Figure 5 (upper part) illustrates the network architecture of TSP-FCOS, with the following components:

Backbone and FPN We follow FCOS [30] on the design of the backbone and the Feature Pyramid Network (FPN) [18]. At the beginning of the pipeline, a backbone CNN is used to extract features from the input images. Based on the feature maps from the backbone, we build the FPN component which produces multi-level features that can help encoder-only DETR detect objects of various scales.

Feature extraction subnets For a fair comparison with other one-stage detectors (*e.g.*, FCOS and RetinaNet), we follow their design and use two feature extraction heads shared across different feature pyramid levels. We call one of them classification subnet (head), which is used for FoI classification. The other is called auxiliary subnet (head). Their outputs are concatenated and then selected by FoI classifier.

Feature of Interest (FoI) classifier In the self-attention module of Transformer, the computation complexity is quadratic to the sequence length, which prohibits directly using all the features on the feature pyramids. To improve the efficiency of self-attention, we design a binary classifier to select a limited portion of features and refer them as Features of Interest (FoI). The binary FoI classifier is trained with FCOS's ground-truth assignment rule[3]. After FoI classification, top scored features are picked as FoIs and fed into the Transformer encoder.

Transformer encoder After the FoI selection step, the input to Transformer encoder is a set of FoIs and their corresponding positional encoding. Inside each layer of Transformer encoder, self-attention is performed to aggregate the information of different FoIs. The outputs of the encoder pass through a shared feed forward network, which predicts the category label (including "no object") and bounding box for each FoI.

Positional encoding Following DETR, we generalize the positional encoding of Transformer [32] to the 2D image scenario. Specifically, for a feature point with normalized position $(x, y) \in [0, 1]^2$, its positional encoding is defined as $[PE(x) : PE(y)]$, where $[:]$ denotes concatenation and function PE is defined by:

$$\begin{aligned} PE(x)_{2i} &= \sin(x/10000^{2i/\mathrm{d}_{\text{model}}}) \\ PE(x)_{2i+1} &= \cos(x/10000^{2i/\mathrm{d}_{\text{model}}}) \end{aligned} \tag{3}$$

where $\mathrm{d}_{\text{model}}$ is the dimension of the FoIs.

Faster set prediction training As mentioned in Section 2.2, the object detection task can be viewed as a set prediction problem. Given the set of detection results and

[3]Please refer to the FCOS paper [30] for more details.

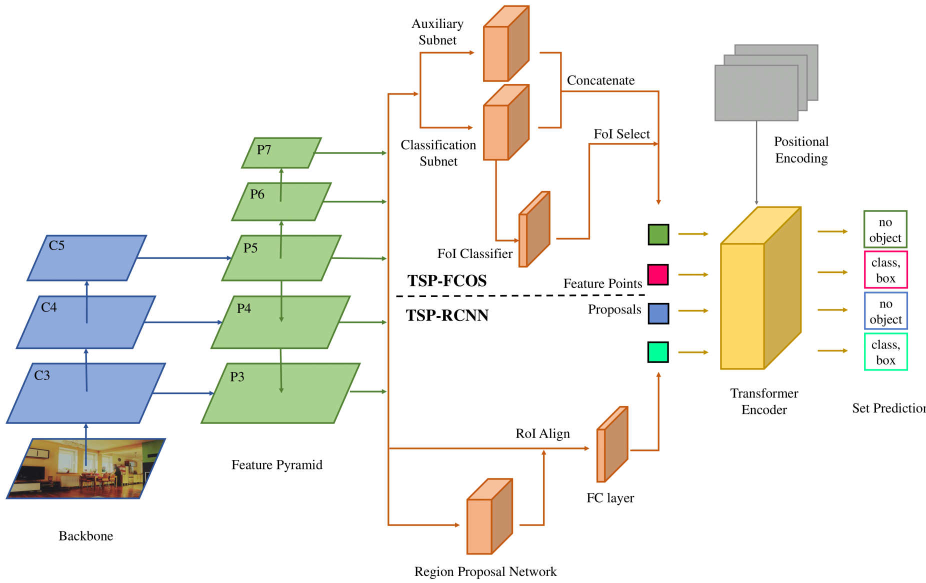

图5. TSP-FCOS与TSP-RCNN的网络架构，其中C_3至C_5表示骨干网络的特征图，P_3至P_7为特征金字塔网络（FPN）的特征图。TSP-FCOS（上图）和TSP-RCNN（下图）均配备Transformer编码器，并通过集合预测损失进行训练。两者的区别在于：TSP-FCOS中的FoI分类器仅预测各特征的目标性（*i.e*，即兴趣特征），而TSP-RCNN中的区域提议网络（RPN）则同时预测边界框及其目标性作为兴趣区域（RoI），即提议框。

兴趣目标（FoI）选择机制，使Transformer编码器能够处理多层次特征，以及一种新的二分匹配方案以加速集合预测训练。图5（上部）展示了TSP-FCOS的网络架构，包含以下组件：

主干网络与FPN 我们遵循FCOS[30]的设计方案构建主干网络和特征金字塔网络(FPN)[18]。在流程起始阶段，主干CNN网络负责从输入图像中提取特征。基于主干网络输出的特征图，我们构建了FPN组件，该组件能生成多层级特征，有助于纯编码器结构的DETR检测不同尺度的目标物体。公式标记{v*}保持不变。

特征提取子网络 为了与其他单阶段检测器（如e.g.、FCOS和RetinaNet）进行公平比较，我们遵循其设计，采用两个共享于不同特征金字塔层级的特征提取头。其中一个称为分类子网络（头），用于兴趣区域分类；另一个称为辅助子网络（头）。两者的输出经拼接后，由兴趣区域分类器进行筛选。

兴趣特征（FoI）分类器 在Transformer的自注意力模块中，计算复杂度与序列长度呈平方关系，这阻碍了直接使用特征金字塔上的所有特征。为了提高自注意力的效率，我们设计了一个二元分类器来筛选有限数量的特征，并将其称为兴趣特征（FoI）。该二元FoI分类器经过训练采用FCOS的真实标注分配规则[3]。在兴趣区域分类后，得分最高的特征被选为兴趣区域并输入到Transformer编码器中。

Transformer编码器 在完成兴趣区域（FoI）选择步骤后，Transformer编码器的输入是一组FoI及其对应的位置编码。在Transformer编码器的每一层内部，通过自注意力机制来聚合不同FoI的信息。编码器的输出经过一个共享的前馈神经网络，该网络为每个FoI预测类别标签（包括“无物体”）和边界框。

位置编码 遵循DETR的做法，我们将Transformer[32]的位置编码推广至二维图像场景。具体而言，对于归一化位置$(x, y) \in [0, 1]^2$的特征点，其位置编码定义为$[PE(x) : PE(y)]$，其中$[:]$表示拼接操作，函数PE的定义如下：

$$\begin{aligned} PE(x)_{2i} &= \sin(x/10000^{2i/\mathrm{d}_{\text{model}}}) \\ PE(x)_{2i+1} &= \cos(x/10000^{2i/\mathrm{d}_{\text{model}}}) \end{aligned} \quad (3)$$

其中$\mathrm{d}_{\text{model}}$是FoIs的维度。

更快的集合预测训练 如2.2节所述，目标检测任务可视为集合预测问题。给定检测结果集合与

[3] Please refer to the FCOS paper [30] for more details.

ground truth objects, the set prediction loss links them together and provides an objective for the model to optimize. But as we show in Section 3.1, the Hungarian bipartite-matching loss can lead to slow convergence in the early stage of training. Therefore, we design a new bipartite matching scheme for faster set prediction training of TSP-FCOS. Specifically, a feature point can be assigned to a ground-truth object only when the point is in the bounding box of the object and in the proper feature pyramid level. This is inspired by the ground-truth assignment rule of FCOS [30]. Next, a restricted cost-based matching process (Equation 2) is performed to determine the optimal matching between the detection results and the ground truth objects in the Hungarian loss (Equation 1).

4.2. TSP-RCNN

Based on the design of TSP-FCOS and Faster RCNN, we can combine the best of them and perform a two-stage bounding box refinement as set prediction, which requires more computational resources but can detect objects more accurately. This idea leads to TSP-RCNN (Transformer-based Set Prediction with RCNN). Figure 5 (lower part) illustrates the network architecture of our proposed TSP-RCNN. The main differences between TSP-FCOS and TSP-RCNN are as follows:

Region proposal network In TSP-RCNN, instead of using two feature extraction heads and FoI classifier to obtain the input of Transformer encoder, we follow the design of Faster RCNN [26] and use a Region Proposal Network (RPN) to get a set of Regions of Interest (RoIs) to be further refined. Different from FoIs in TSP-FCOS, each RoI in TSP-RCNN contains not only an objectness score, but also a predicted bounding box. We apply RoIAlign [11] to extract the information of RoIs from multi-level feature maps. The extracted features are then flattened and fed into a fully connected network as the input of Transformer encoder.

Positional encoding The positional information of a RoI (proposal) is defined by four quantities (cx, cy, w, h), where $(cx, cy) \in [0, 1]^2$ denotes the normalized center coordinates and $(w, h) \in [0, 1]^2$ denotes the normalized height and width. We use $[PE(cx) : PE(cy) : PE(w) : PE(h)]$ as the positional encoding of the proposal, where PE and $[:]$ is defined in the same way as TSP-FCOS.

Faster set prediction training TSP-RCNN is also trained with a set prediction loss. Different from TSP-FCOS, we borrow the ground-truth assignment rule from Faster RCNN for faster set prediction training of TSP-RCNN. Specifically, a proposal can be assigned to a ground-truth object if and only if the intersection-over-union (IoU) score between their bounding boxes is greater than 0.5.

5. Experiments

5.1. Dataset and Evaluation Metrics

We evaluate our methods on the COCO [20] object detection dataset, which includes 80 object classes. Following the common practice [19, 30], we use all $115k$ images in `trainval35k` split for training and all $5k$ images in `minival` split for validation. The test result is obtained by submitting the results of `test-dev` split to the evaluation server. For comparison with other methods, we mainly focus on the Average Precision (AP), which is the primary challenge metric used in COCO, and FLOPs, which measures the computation overhead.

5.2. Implementation Details

We briefly describe the default settings of our implementation. More detailed settings can be found in appendix.

TSP-FCOS Following FCOS [30], both classification subnet and auxiliary subnet use four 3×3 convolutional layers with 256 channels and group normalization [33]. In FoI selection, we select top 700 scored feature positions from FoI classifier as the input of Transformer encoder.

TSP-RCNN Different from the original Faster RCNN, we apply 2 unshared convolutional subnets to P_3-P_7 as classification and regression heads of RPN and use a RetinaNet [19] style anchor generation scheme. We find this improves the performance of RPN with less computation overhead. In RoI selection, we select top 700 scored features from RPN. RoI Align operation [11] and a fully connected layer are applied to extract the proposal features from RoIs.

Transformer encoder As both TSP-FCOS and TSP-RNN only have a Transformer encoder while DETR has both Transformer encoder and decoder, to be comparable in terms of FLOPs with DETR-DC5, we use a 6-layer Transformer encoder of width 512 with 8 attention heads. The hidden size of feed-forward network (FFN) in Transformer is set to 2048. During training, we randomly drop 70% inputs of Transformer encoder to improve the robustness of set prediction.

Training We follow the default setting of Detectron2 [34], where a 36-epoch ($3\times$) schedule with multi-scale train-time augmentation is used.

5.3. Main Results

Table 1 shows our main results on COCO 2017 validation set. We compare TSP-FCOS and TSP-RCNN with FCOS [30], Faster RCNN [26], and DETR [2]. We also compare with concurrent work on improving DETR: Deformable DETR [42] and UP-DETR [8]. From the table,

真实标注对象，集合预测损失将它们联系在一起，并为模型提供了优化的目标。但正如我们在第3.1节所示，匈牙利二分图匹配损失可能导致训练初期收敛速度较慢。因此，我们设计了一种新的二分图匹配方案，以加速TSP-FCOS的集合预测训练。具体而言，只有当特征点位于对象的边界框内且处于适当的特征金字塔层级时，该点才能被分配给一个真实标注对象。这一设计灵感来源于FCOS[30]的真实标注分配规则。接着，执行一个基于受限成本的匹配过程（公式2），以在匈牙利损失（公式1）中确定检测结果与真实标注对象之间的最优匹配。

4.2. TSP-RCNN

基于TSP-FCOS与Faster RCNN的设计理念，我们可以融合二者的优势，采用两阶段边界框精细化作为集合预测，虽然需要更多计算资源，但能实现更精确的目标检测。这一思路催生了TSP-RCNN（基于Transformer与RCNN的集合预测模型）。图5（下半部分）展示了我们提出的TSP-RCNN网络架构。TSP-FCOS与TSP-RCNN的主要区别如下:

区域提议网络 在TSP-RCNN中，我们并未采用双特征提取头与FoI分类器来获取Transformer编码器的输入，而是沿用了Faster RCNN[26]的设计，通过区域提议网络（RPN）生成一组待优化的感兴趣区域（RoIs）。与TSP-FCOS中的FoIs不同，TSP-RCNN中的每个RoI不仅包含目标性评分，还带有预测边界框信息。我们采用RoIAlign[11]技术从多层次特征图中提取RoIs的特征信息，随后将这些特征展平并输入全连接网络，作为Transformer编码器的输入。

位置编码 一个RoI（提案）的位置信息由四个量（cx, cy, w, h）定义，其中（cx, cy）$\in [0, 1]^2$表示归一化的中心坐标，（w, h）$\in [0, 1]^2$表示归一化的高度和宽度。我们使用$[PE(cx) : PE(cy) : PE(w) : PE(h)]$作为该提案的位置编码，其中$PE$和[:]的定义方式与TSP-FCOS相同。

更快的集合预测训练 TSP-RCNN同样采用集合预测损失进行训练。与TSP-FCOS不同，我们借鉴了Faster RCNN的真实标注分配规则，以加速TSP-RCNN的集合预测训练。具体而言，当且仅当提案框与真实标注框的交并比（IoU）得分大于0.5时，该提案框才会被分配给对应的真实标注对象。

5. 实验

5.1. 数据集与评估指标

我们在COCO[20]目标检测数据集上评估了我们的方法，该数据集包含80个目标类别。遵循通用实践[19,30]，我们使用trainval35k分割中的所有115k张图像进行训练，并使用minival分割中的所有5k张图像进行验证。测试结果通过将test-dev分割的预测结果提交至评估服务器获得。为与其他方法进行比较，我们主要关注平均精度（AP）——这是COCO采用的核心挑战指标，以及衡量计算开销的浮点运算次数（FLOPs）。

5.2. 实现细节

我们简要描述了我们实现的默认设置。更详细的设置可在附录中找到。

TSP-FCOS 遵循FCOS[30]的设计，分类子网和辅助子网均采用四层3×3卷积结构，每层包含256个通道并应用组归一化[33]。在兴趣区域选择阶段，我们从兴趣区域分类器中筛选出得分最高的700个特征位置，作为Transformer编码器的输入。

TSP-RCNN 与原始Faster RCNN不同，我们在P_3-P_7上应用了两个不共享参数的卷积子网络，作为RPN的分类和回归头，并采用了RetinaNet[19]风格的锚框生成方案。我们发现这种做法能以更少的计算开销提升RPN的性能。在候选区域选择阶段，我们从RPN中筛选出得分最高的700个特征。通过RoI Align操作[11]和全连接层，从候选区域中提取出提案特征。

Transformer编码器 由于TSP-FCOS和TSP-RNN仅包含Transformer编码器，而DETR同时具备Transformer编码器和解码器，为了在FLOPs上与DETR-DC5具有可比性，我们采用了宽度为512、8个注意力头的6层Transformer编码器。Transformer中前馈网络（FFN）的隐藏层大小设置为2048。训练过程中，我们随机丢弃70%的Transformer编码器输入以提升集合预测的鲁棒性。

训练 我们遵循Detectron2 [34]的默认设置，采用36个周期（3{v*}）的多尺度训练时增强计划。

5.3. 主要结果

表1展示了我们在COCO 2017验证集上的主要结果。我们将TSP-FCOS和TSP-RCNN与FCOS[30]、Faster RCNN[26]以及DETR[2]进行了比较。同时，我们还对比了同期关于改进DETR的研究工作：可变形DETR[42]和UP-DETR[8]。从表中可见，

Model	Backbone	Epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	FLOPs	FPS
FCOS†	ResNet-50	36	41.0	59.8	44.1	26.2	44.6	52.2	177G	17
Faster RCNN-FPN	ResNet-50	36	40.2	61.0	43.8	24.2	43.5	52.0	180G	19
Faster RCNN-FPN+	ResNet-50	108	42.0	62.1	45.5	26.6	45.4	53.4	180G	19
DETR+	ResNet-50	500	42.0	62.4	44.2	20.5	45.8	61.1	86G	21
DETR-DC5+	ResNet-50	500	43.3	63.1	45.9	22.5	47.3	61.1	187G	7
Deformable DETR*	ResNet-50	50	43.8	62.6	47.7	26.4	47.1	58.0	173G	-
UP-DETR	ResNet-50	300	42.8	63.0	45.3	20.8	47.1	**61.7**	86G	21
TSP-FCOS	ResNet-50	36	43.1	62.3	47.0	26.6	46.8	55.9	189G	15
TSP-RCNN	ResNet-50	36	43.8	63.3	48.3	28.6	46.9	55.7	188G	11
TSP-RCNN+	ResNet-50	96	**45.0**	**64.5**	**49.6**	**29.7**	**47.7**	58.0	188G	11
FCOS†	ResNet-101	36	42.5	61.3	45.9	26.0	46.5	53.6	243G	13
Faster RCNN-FPN	ResNet-101	36	42.0	62.5	45.9	25.2	45.6	54.6	246G	15
Faster RCNN-FPN+	ResNet-101	108	44.0	63.9	47.8	27.2	48.1	56.0	246G	15
DETR+	ResNet-101	500	43.5	63.8	46.4	21.9	48.0	61.8	152G	15
DETR-DC5+	ResNet-101	500	44.9	64.7	47.7	23.7	49.5	**62.3**	253G	6
TSP-FCOS	ResNet-101	36	44.4	63.8	48.2	27.7	48.6	57.3	255G	12
TSP-RCNN	ResNet-101	36	44.8	63.8	49.2	29.0	47.9	57.1	254G	9
TSP-RCNN+	ResNet-101	96	**46.5**	**66.0**	**51.2**	**29.9**	**49.7**	59.2	254G	9

Table 1. Evaluation results on COCO 2017 validation set. † represents our reproduction results. + represents that the models are trained with random crop augmentation and a longer training schedule. We use Detectron2 package to measure FLOPs and FPS. A single Nvidia GeForce RTX 2080 Ti GPU is used for measuring inference latency. ∗ represents the version without iterative refinement. A fair comparison of TSP-RCNN and Deformabel DETR both with iterative refinement can be found in the appendix.

Model	AP	AP_S	AP_M	AP_L
TSP-RCNN-R50	**43.8**	**28.6**	**46.9**	55.7
w/o set prediction loss	42.7	27.6	45.5	**56.2**
w/o positional encoding	43.4	28.4	46.3	55.0
TSP-RCNN-R101	**44.8**	**29.0**	**47.9**	**57.1**
w/o set prediction loss	44.0	27.6	47.2	**57.1**
w/o positional encoding	44.4	28.2	47.7	56.7

Table 2. Evaluation results on COCO 2017 validation set for ablation study of set prediction loss and positional encoding.

Model	AP	AP_S	AP_M	AP_L
FCOS	45.3	28.1	49.0	59.3
TSP-FCOS	**46.1**	**28.5**	**49.7**	**60.2**
Faster-RCNN	44.1	26.4	47.6	58.1
TSP-RCNN	**45.8**	**29.4**	**49.2**	**58.4**

Table 3. Evaluation results on COCO 2017 validation set with ResNet-101-DCN backbone.

we can see that our TSP-FCOS and TSP-RCNN significantly outperform original FCOS and Faster RCNN. Besides, we can find that TSP-RCNN is better than TSP-FCOS in terms of overall performance and small object detection but slightly worse in terms of inference latency.

To compare with state-of-the-art DETR models, we use a similar training strategy in DETR [2], where a 96-epoch (8×) training schedule and random crop augmentation is applied. We denote the enhanced version of TSP-RCNN by TSP-RCNN+. We also copy the results of enhanced Faster RCNN (*i.e.*, Faster RCNN+) from [2]. Comparing these models, we can find that our TSP-RCNN obtains state-of-the-art results with a shorter training schedule. We also find that TSP-RCNN+ still under-performs DETR-DC5+ on large object detection. We think this is because of the inductive bias of the encoder-decoder scheme used by DETR and its longer training schedule.

5.4. Model Analysis

For model analysis, we evaluate several models trained in our default setting, *i.e.*, with a 36-epoch (3×) schedule and without random crop augmentation.

5.4.1 Ablation study

We conduct an ablation study of set prediction loss and positional encoding, which are two essential components in our model. Table 2 show the results of ablation study for TSP-RCNN with ResNet-50 and ResNet-101 backbone. From the table, we can see that both set prediction loss and positional encoding are very important to the success of our TSP mechanism, while set prediction loss contributes more than positional encoding to the improvement of TSP-RCNN.

Model	Backbone	Epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	FLOPs	FPS
FCOS†	ResNet-50	36	41.0	59.8	44.1	26.2	44.6	52.2	177G	17
Faster RCNN-FPN	ResNet-50	36	40.2	61.0	43.8	24.2	43.5	52.0	180G	19
Faster RCNN-FPN+	ResNet-50	108	42.0	62.1	45.5	26.6	45.4	53.4	180G	19
DETR+	ResNet-50	500	42.0	62.4	44.2	20.5	45.8	61.1	86G	21
DETR-DC5+	ResNet-50	500	43.3	63.1	45.9	22.5	47.3	61.1	187G	7
Deformable DETR*	ResNet-50	50	43.8	62.6	47.7	26.4	47.1	58.0	173G	-
UP-DETR	ResNet-50	300	42.8	63.0	45.3	20.8	47.1	**61.7**	86G	21
TSP-FCOS	ResNet-50	36	43.1	62.3	47.0	26.6	46.8	55.9	189G	15
TSP-RCNN	ResNet-50	36	43.8	63.3	48.3	28.6	46.9	55.7	188G	11
TSP-RCNN+	ResNet-50	96	**45.0**	**64.5**	**49.6**	**29.7**	**47.7**	58.0	188G	11
FCOS†	ResNet-101	36	42.5	61.3	45.9	26.0	46.5	53.6	243G	13
Faster RCNN-FPN	ResNet-101	36	42.0	62.5	45.9	25.2	45.6	54.6	246G	15
Faster RCNN-FPN+	ResNet-101	108	44.0	63.9	47.8	27.2	48.1	56.0	246G	15
DETR+	ResNet-101	500	43.5	63.8	46.4	21.9	48.0	61.8	152G	15
DETR-DC5+	ResNet-101	500	44.9	64.7	47.7	23.7	49.5	**62.3**	253G	6
TSP-FCOS	ResNet-101	36	44.4	63.8	48.2	27.7	48.6	57.3	255G	12
TSP-RCNN	ResNet-101	36	44.8	63.8	49.2	29.0	47.9	57.1	254G	9
TSP-RCNN+	ResNet-101	96	**46.5**	**66.0**	**51.2**	**29.9**	**49.7**	59.2	254G	9

表1. COCO 2017验证集上的评估结果。†代表我们的复现结果。+表示模型采用了随机裁剪增强和更长的训练周期。我们使用Detectron2包来测量FLOPs和FPS，并采用单块Nvidia GeForce RTX 2080 Ti GPU进行推理延迟测试。∗代表未使用迭代优化的版本。TSP-RCNN与Deformable DETR在均采用迭代优化条件下的公平对比结果详见附录。

Model	AP	AP_S	AP_M	AP_L
TSP-RCNN-R50	**43.8**	**28.6**	**46.9**	55.7
w/o set prediction loss	42.7	27.6	45.5	**56.2**
w/o positional encoding	43.4	28.4	46.3	55.0
TSP-RCNN-R101	**44.8**	**29.0**	**47.9**	**57.1**
w/o set prediction loss	44.0	27.6	47.2	**57.1**
w/o positional encoding	44.4	28.2	47.7	56.7

表2. 在COCO 2017验证集上关于集合预测损失与位置编码消融研究的评估结果。

Model	AP	AP_S	AP_M	AP_L
FCOS	45.3	28.1	49.0	59.3
TSP-FCOS	**46.1**	**28.5**	**49.7**	**60.2**
Faster-RCNN	44.1	26.4	47.6	58.1
TSP-RCNN	**45.8**	**29.4**	**49.2**	**58.4**

表3. 使用ResNet-101-DCN骨干网络在COCO 2017验证集上的评估结果。

可以看出，我们的TSP-FCOS和TSP-RCNN显著优于原始FCOS与Faster RCNN。此外，TSP-RCNN在整体性能和小物体检测方面优于TSP-FCOS，但在推理延迟方面稍逊一筹。

为了与最先进的DETR模型进行比较，我们采用了DETR[2]中类似的训练策略，即使用96个epoch的(8×)的训练计划并应用了随机裁剪增强。我们将增强版的TSP-RCNN记为TSP-RCNN+。同时，我们从文献[2]中复制了增强版Faster RCNN（*i.e*，即Faster RCNN+）的结果。通过比较这些模型可以发现，我们的TSP-RCNN以更短的训练周期取得了最先进的结果。我们还发现，TSP-RCNN+在大物体检测上仍逊色于DETR-DC5+。我们认为这是由于DETR采用的编码器-解码器结构所带来的归纳偏置及其更长的训练周期所致。

5.4. 模型分析

对于模型分析，我们评估了在默认设置*i.e*下训练的多个模型，采用36周期（3×）的训练计划，且未使用随机裁剪增强。

5.4.1 消融研究

我们对集合预测损失和位置编码进行了消融研究，这两者是我们模型中的两个关键组成部分。表2展示了基于ResNet-50和ResNet-101骨干网络的TSP-RCNN消融实验结果。从表中可以看出，集合预测损失和位置编码对于TSP机制的成功都非常重要，而集合预测损失对TSP-RCNN性能提升的贡献大于位置编码。

Model	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
RetinaNet [19]	ResNet-101	39.1	59.1	42.3	21.8	42.7	50.2
FSAF [40]	ResNet-101	40.9	61.5	44.0	24.0	44.2	51.3
FCOS [30]	ResNet-101	41.5	60.7	45.0	24.4	44.8	51.6
MAL [15]	ResNet-101	43.6	62.8	47.1	25.0	46.9	55.8
RepPoints [35]	ResNet-101-DCN	45.0	66.1	49.0	26.6	48.6	57.5
ATSS [38]	ResNet-101	43.6	62.1	47.4	26.1	47.0	53.6
ATSS [38]	ResNet-101-DCN	46.3	64.7	50.4	27.7	49.8	58.4
Fitness NMS [31]	ResNet-101	41.8	60.9	44.9	21.5	45.0	57.5
Libra RCNN [22]	ResNet-101	41.1	62.1	44.7	23.4	43.7	52.5
Cascade RCNN [1]	ResNet-101	42.8	62.1	46.3	23.7	45.5	55.2
TridentNet [17]	ResNet-101-DCN	46.8	**67.6**	51.5	28.0	**51.2**	**60.5**
TSD [28]	ResNet-101	43.2	64.0	46.9	24.0	46.3	55.8
Dynamic RCNN [37]	ResNet-101	44.7	63.6	49.1	26.0	47.4	57.2
Dynamic RCNN [37]	ResNet-101-DCN	46.9	65.9	51.3	28.1	49.6	60.0
TSP-RCNN	ResNet-101	46.6	66.2	51.3	28.4	49.0	58.5
TSP-RCNN	ResNet-101-DCN	**47.4**	66.7	**51.9**	**29.0**	49.7	59.1

Table 4. Comparison with state-of-the-art models on COCO 2017 test set (single-model and single-scale results). Underlined and **bold** numbers represent the best model with ResNet-101 and ResNet-101-DCN backbone, respectively.

5.4.2 Compatibility with deformable convolutions

One may wonder whether Transformer encoder and deformable convolutions [7, 41] are compatible with each other, as both of them can utilize long-range relation between objects. In Table 3, we compare TSP-FCOS and TSP-RCNN to FCOS and Faster RCNN with deformable ResNet-101 as backbone. From the results, we can see that the TSP mechanism is well complementary with deformable convolutions.

5.5. Comparison with State-of-the-Arts

We compare TSP-RCNN with multiple one-stage and two-stage object detection models [26, 31, 1, 28, 19, 40, 30, 3, 16, 35, 38] that also use ResNet-101 backbone or its deformable convolution network (DCN) [41] variant in Table 4. A $8\times$ schedule and random crop augmentation is used. The performance metrics are evaluated on COCO 2017 test set using single-model and single-scale detection results. Our model achieves the highest AP scores among all detectors in both backbone settings.

6. Analysis of convergence

We compare the convergence speed of our faster set prediction training and DETR's original set prediction training in the upper part of Figure 6. We can see that our proposed faster training technique consistently accelerates the convergence of both TSP-FCOS and TSP-RCNN.

We also plot the convergence curves of TSP-FCOS, TSP-RCNN, and DETR-DC5 in the lower part of Figure 6, from which we can find that our proposed models not only converge faster, but also achieve better detection performance.

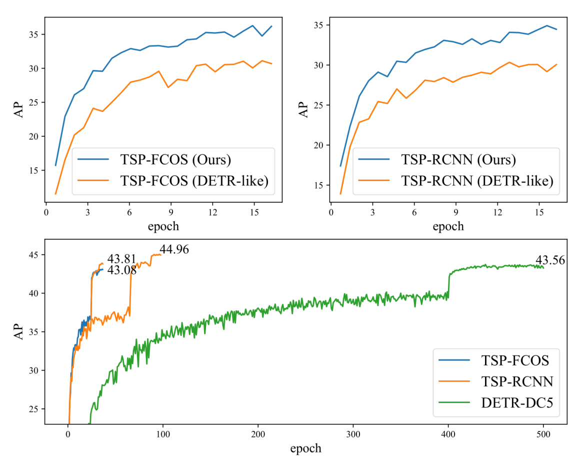

Figure 6. Top two plots compare the convergence speed of our proposed faster set prediction training loss and DETR-like loss for TSP-FCOS and TSP-RCNN. The bottom plot shows the convergence curves of TSP-FCOS, TSP-RCNN, and DETR-DC5.

7. Conclusion

Aiming to accelerate the training convergence of DETR as well as to improve prediction power in object detection, we present an investigation on the causes of its slow convergence through extensive experiments, and propose two novel solutions, namely TSP-FCOS and TSP-RCNN, which require much less training time and achieve the state-of-the-art detection performance. For future work, we would like to investigate the successful use of sparse attention mechanism [5, 6, 36] for directly modeling the relationship among multi-level features.

Model	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
RetinaNet [19]	ResNet-101	39.1	59.1	42.3	21.8	42.7	50.2
FSAF [40]	ResNet-101	40.9	61.5	44.0	24.0	44.2	51.3
FCOS [30]	ResNet-101	41.5	60.7	45.0	24.4	44.8	51.6
MAL [15]	ResNet-101	43.6	62.8	47.1	25.0	46.9	55.8
RepPoints [35]	ResNet-101-DCN	45.0	66.1	49.0	26.6	48.6	57.5
ATSS [38]	ResNet-101	43.6	62.1	47.4	26.1	47.0	53.6
ATSS [38]	ResNet-101-DCN	46.3	64.7	50.4	27.7	49.8	58.4
Fitness NMS [31]	ResNet-101	41.8	60.9	44.9	21.5	45.0	57.5
Libra RCNN [22]	ResNet-101	41.1	62.1	44.7	23.4	43.7	52.5
Cascade RCNN [1]	ResNet-101	42.8	62.1	46.3	23.7	45.5	55.2
TridentNet [17]	ResNet-101-DCN	46.8	**67.6**	51.5	28.0	**51.2**	**60.5**
TSD [28]	ResNet-101	43.2	64.0	46.9	24.0	46.3	55.8
Dynamic RCNN [37]	ResNet-101	44.7	63.6	49.1	26.0	47.4	57.2
Dynamic RCNN [37]	ResNet-101-DCN	46.9	65.9	51.3	28.1	49.6	60.0
TSP-RCNN	ResNet-101	46.6	66.2	51.3	28.4	49.0	58.5
TSP-RCNN	ResNet-101-DCN	**47.4**	66.7	**51.9**	**29.0**	49.7	59.1

表4. 在COCO 2017测试集上与最先进模型的比较（单模型和单尺度结果）。下划线和加粗数字分别代表以ResNet-101和ResNet-101-DCN为骨干网络的最佳模型。

5.4.2 与可变形卷积的兼容性

人们可能会好奇Transformer编码器与可变形卷积[7,41]是否彼此兼容，因为两者都能利用物体间的长程关系。在表3中，我们将TSP-FCOS和TSP-RCNN与采用可变形ResNet-101作为骨干网络的FCOS及Faster RCNN进行对比。从结果可以看出，TSP机制与可变形卷积具有很好的互补性。

5.5. 与最先进技术的比较

我们将TSP-RCNN与多种单阶段和两阶段目标检测模型[26, 31, 1, 28, 19, 40, 30, 3, 16, 35, 38]进行对比，这些模型同样采用ResNet-101主干网络或其可变形卷积网络(DCN)[41]变体，对比结果如表4所示。实验采用8×训练周期和随机裁剪数据增强策略，性能指标基于COCO 2017测试集的单模型单尺度检测结果进行评估。在两种主干网络配置下，我们的模型均实现了所有检测器中最高的AP分数。

6. 收敛性分析

在图6的上半部分，我们比较了更快的集合预测训练与DETR原始集合预测训练的收敛速度。可以看出，我们提出的快速训练技术持续加速了TSP-FCOS和TSP-RCNN的收敛过程。

我们还在图6的下半部分绘制了TSP-FCOS、TSP-RCNN和DETR-DC5的收敛曲线，从中可以发现，我们提出的模型不仅收敛速度更快，而且实现了更优的检测性能。

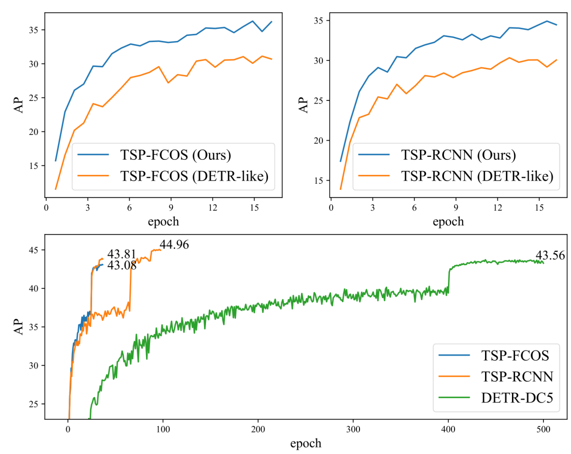

图6. 上方两幅图比较了我们提出的更快集合预测训练损失与类似DETR损失在TSP-FCOS和TSP-RCNN上的收敛速度。底部图展示了TSP-FCOS、TSP-RCNN和DETR-DC5的收敛曲线。

7. 结论

为了加速DETR的训练收敛并提升其在目标检测中的预测能力，我们通过大量实验探究了其收敛缓慢的原因，并提出了两种创新解决方案——TSP-FCOS与TSP-RCNN。这些方案大幅减少了训练时间，同时实现了最先进的检测性能。未来工作中，我们将探索如何成功运用稀疏注意力机制{v*}，直接建模多层级特征间的关系。

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018. 8

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2020. 1, 2, 3, 6, 7

[3] Yuntao Chen, Chenxia Han, Naiyan Wang, and Zhaoxiang Zhang. Revisiting feature alignment for one-stage object detection. *arXiv preprint arXiv:1908.01570*, 2019. 8

[4] Cheng Chi, Fangyun Wei, and Han Hu. Relationnet++: Bridging visual representations for object detection via transformer decoder. *Advances in Neural Information Processing Systems*, 33, 2020. 2

[5] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019. 8

[6] Gonçalo M Correia, Vlad Niculae, and André FT Martins. Adaptively sparse transformers. *arXiv preprint arXiv:1909.00015*, 2019. 8

[7] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 8

[8] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. *arXiv preprint arXiv:2011.09094*, 2020. 2, 6

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2, 3

[10] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 2

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1, 2, 6

[12] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3588–3597, 2018. 2

[13] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 3

[14] Zi-Hang Jiang, Weihao Yu, Daquan Zhou, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. Convbert: Improving bert with span-based dynamic convolution. *Advances in Neural Information Processing Systems*, 33, 2020. 3

[15] Wei Ke, Tianliang Zhang, Zeyi Huang, Qixiang Ye, Jianzhuang Liu, and Dong Huang. Multiple anchor learning for visual object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10206–10215, 2020. 8

[16] Tao Kong, Fuchun Sun, Huaping Liu, Yuning Jiang, Lei Li, and Jianbo Shi. Foveabox: Beyound anchor-based object detection. *IEEE Transactions on Image Processing*, 29:7389–7398, 2020. 8

[17] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 6054–6063, 2019. 8

[18] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 1, 4, 5

[19] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 1, 2, 6, 8

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1, 4, 6

[21] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 1, 2

[22] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra r-cnn: Towards balanced learning for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 821–830, 2019. 8

[23] Zheng Qin, Zeming Li, Zhaoning Zhang, Yiping Bao, Gang Yu, Yuxing Peng, and Jian Sun. Thundernet: Towards real-time generic object detection on mobile devices. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6718–6727, 2019. 2

[24] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1, 2

[25] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 1

[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1, 2, 6, 8

[27] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 658–666, 2019. 2

[28] Guanglu Song, Yu Liu, and Xiaogang Wang. Revisiting the sibling head in object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11563–11572, 2020. 8

参考文献

[1] 蔡兆伟与Nuno Vasconcelos。级联R-CNN：深入高质量目标检测。于*Proceedings of the IEEE conference on computer vision and pattern recogni-tion*，页码6154–6162，2018年。8 [2] Nicolas Carion、Francisco Massa、Gabriel Synnaeve、Nicolas Usunier、Alexander Kirillov及Sergey Zagoruyko。基于Transformer的端到端目标检测。*arXiv preprintarXiv:2005.12872*，2020年。1, 2, 3, 6, 7 [3] 陈运涛、韩晨曦、王乃岩与张兆翔。重新审视单阶段目标检测中的特征对齐。*arXiv preprint arXiv:1908.01570*，2019年。8 [4] 迟程、魏芳云与胡涵。RelationNet++：通过Transformer解码器桥接目标检测的视觉表示。*Advances in Neural Information ProcessingSystems*，33卷，2020年。2 [5] Rewon Child、Scott Gray、Alec Radford与Ilya Sutskever。利用稀疏Transformer生成长序列。*arXiv preprint arXiv:1904.10509*，2019年。8 [6] Gonçalo M Correia、Vlad Niculae与André FT Martins。自适应稀疏Transformer。*arXiv preprintarXiv:1909.00015*，2019年。8 [7] 戴继峰、齐浩志、熊宇闻、李毅、张国栋、胡涵与魏亦宸。可变形卷积网络。载于*Proceedings of the IEEE international confer-ence on computer vision*，页码764–773，2017年。8 [8] 戴志刚、蔡博伦、林雨耕与陈俊颖。UP-DETR：基于Transformer的目标检测无监督预训练。*arXiv preprint arXiv:2011.09094*，2020年。2, 6 [9] Jacob Devlin、张梦伟、Kenton Lee与Kristina Toutanova。BERT：面向语言理解的深度双向Transformer预训练。*arXiv preprintarXiv:1810.04805*，2018年。2, 3 [10] Ross Girshick。Fast R-CNN。载于*Proceedings of the IEEE inter-national conference on computer vision*，页码1440–1448，2015年。2 [11] 何恺明、Georgia Gkioxari、Piotr Dollár与Ross Girshick。Mask R-CNN。载于*Proceedings of the IEEE international conference on computer vision*，页码2961–2969，2017年。1, 2, 6 [12] 胡涵、顾家圆、张政、戴继峰与魏亦宸。目标检测中的关系网络。载于*Proceed-ings of the IEEE Conference on Computer Vision and Pattern Recognition*，页码3588–3597，2018年。2 [13] Sergey Ioffe与Christian Szegedy。批量归一化：通过减少内部协变量偏移加速深度网络训练。*arXiv preprint arXiv:1502.03167*，2015年。3 [14] 姜子航、余炜皓、周达权、陈云鹏、冯佳时与颜水成。ConvBERT：利用基于跨度的动态卷积改进BERT。*Advances in NeuralInformation Processing Systems*，33卷，2020年。3 [15] 柯炜、张天良、黄泽毅、叶启翔、刘建柱与黄栋。视觉目标检测的多锚点学习。载于*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*，页码10206–10215，2020年。8

[16] 陶孔，孙富春，刘华平，蒋宇宁，李磊，史建波。FoveaBox：超越基于锚点的目标检测。*IEEE Transactions on Image Processing*，29:7389–7398，2020年。8 [17] 李阳浩，陈云涛，王乃岩，张兆翔。面向目标检测的尺度感知三叉戟网络。载于*Proceedings of the IEEE international conference on com-puter vision*，页6054–6063，2019年。8 [18] 林腾毅，Piotr Dollár，Ross Girshick，何恺明，Bharath Hariharan，Serge Belongie。目标检测的特征金字塔网络。载于*Proceedings of the IEEE conference on computer vision and pattern recogni-tion*，页2117–2125，2017年。1, 4, 5 [19] 林腾毅，Priya Goyal，Ross Girshick，何恺明，Piotr Dollár。密集目标检测的焦点损失。载于*Pro-ceedings of the IEEE international conference on computer vision*，页2980–2988，2017年。1, 2, 6, 8 [20] 林腾毅，Michael Maire，Serge Belongie，James Hays，Pietro Perona，Deva Ramanan，Piotr Dollár，C Lawrence Zitnick。Microsoft COCO：上下文中的常见物体。载于*European conference on computer vision*，页740–755。Springer，2014年。1, 4, 6 [21] 刘伟，Dragomir Anguelov，Dumitru Erhan，Christian Szegedy，Scott Reed，傅成阳，Alexander C Berg。SSD：单次多框检测器。载于*European con-ference on computer vision*，页21–37。Springer，2016年。1, 2 [22] 庞江淼，陈凯，史建平，冯华君，欧阳万里，林达华。Libra R-CNN：迈向目标检测的平衡学习。载于*Proceedings of the IEEE conference on computer vision and pattern recogni-tion*，页821–830，2019年。8 [23] 覃正，李子铭，张昭宁，鲍一平，余刚，彭宇星，孙剑。ThunderNet：面向移动设备的实时通用目标检测。载于*Proceed-ings of the IEEE International Conference on Computer Vi-sion*，页6718–6727，2019年。2 [24] Joseph Redmon，Santosh Divvala，Ross Girshick，Ali Farhadi。YOLO：统一实时目标检测。载于*Proceedings of the IEEE conference on computer vision and pattern recognition*，页779–788，2016年。1, 2 [25] Joseph Redmon，Ali Farhadi。YOLO9000：更好、更快、更强。载于*Proceedings of the IEEE conference on computer vision and pattern recognition*，页7263–7271，2017年。1 [26] 任少卿，何恺明，Ross Girshick，孙剑。Faster R-CNN：利用区域提议网络实现实时目标检测。载于*Advances in neural information pro-cessing systems*，页91–99，2015年。1, 2, 6, 8 [27] Hamid Rezatofighi，Nathan Tsoi，JunYoung Gwak，Amir Sadeghian，Ian Reid，Silvio Savarese。广义交并比：边界框回归的度量与损失。载于*Proceedings of the IEEE Conference on Com-puter Vision and Pattern Recognition*，页658–666，2019年。2 [28] 宋广鲁，刘宇，王晓刚。重新审视目标检测中的兄弟头结构。载于*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*，页11563–11572，2020年。8

[29] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 3

[30] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 9627–9636, 2019. 1, 4, 5, 6, 8

[31] Lachlan Tychsen-Smith and Lars Petersson. Improving object localization with fitness nms and bounded iou loss. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6877–6885, 2018. 8

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 1, 2, 5

[33] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 6

[34] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2, 2019. 6

[35] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9657–9666, 2019. 8

[36] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33, 2020. 8

[37] Hongkai Zhang, Hong Chang, Bingpeng Ma, Naiyan Wang, and Xilin Chen. Dynamic r-cnn: Towards high quality object detection via dynamic training. *arXiv preprint arXiv:2004.06002*, 2020. 8

[38] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9759–9768, 2020. 1, 8

[39] Xiaosong Zhang, Fang Wan, Chang Liu, Rongrong Ji, and Qixiang Ye. Freeanchor: Learning to match anchors for visual object detection. In *Advances in Neural Information Processing Systems*, pages 147–155, 2019. 2

[40] Chenchen Zhu, Yihui He, and Marios Savvides. Feature selective anchor-free module for single-shot object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 840–849, 2019. 8

[41] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9308–9316, 2019. 8

[42] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 2, 6

[29] Nitish Srivastava、Geoffrey Hinton、Alex Krizhevsky、Ilya Sutskever 和 Ruslan Salakhutdinov。Dropout：一种防止神经网络过拟合的简单方法。*The journal of machine learning research*，15(1):1929–1958，2014年。3 [30] 田值、沈春华、陈浩和何通。FCOS：全卷积一阶段目标检测。载于*Proceedings of the IEEE international conference on computer vi- sion*，页码9627–9636，2019年。1, 4, 5, 6, 8 [31] Lachlan Tychsen-Smith 和 Lars Petersson。通过适应度NMS和有界IoU损失改进目标定位。载于*Proceedings of the IEEE conference on computer vision and pattern recognition*，页码6877–6885，2018年。8 [32] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、 ukasz Kaiser 和 Illia Polosukhin。注意力机制就是你所需要的一切。载于*Advances in neural information processing systems*，页码5998–6008，2017年。1, 2, 5 [33] 吴育昕和何恺明。组归一化。载于*Proceedings of the European conference on computer vision (ECCV)*，页码3–19，2018年。6 [34] 吴育昕、Alexander Kirillov、Francisco Massa、Wan-Yen Lo 和 Ross Girshick。Detectron2，2019年。6 [35] 杨泽、刘少辉、胡涵、王立伟和林斯蒂芬。RepPoints：用于目标检测的点集表示。载于*Proceedings of the IEEE International Conference on Computer Vision*，页码9657–9666，2019年。8 [36] Manzil Zaheer、Guru Guruganesh、Kumar Avinava Dubey、Joshua Ainslie、Chris Alberti、Santiago Ontanon、Philip Pham、Anirudh Ravula、王启帆、李阳等。Big Bird：面向更长序列的Transformer模型。*Advances in Neural Information Processing Systems*，33卷，2020年。8 [37] 张洪凯、常宏、马兵鹏、王乃岩和陈熙霖。Dynamic R-CNN：通过动态训练实现高质量目标检测。*arXiv preprint arXiv:2004.06002*，2020年。8 [38] 张世峰、迟程、姚永强、雷震和李子斌。通过自适应训练样本选择弥合基于锚点与无锚点检测间的差距。载于*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*，页码9759–9768，2020年。1, 8 [39] 张晓松、万方、刘畅、纪荣荣和叶启祥。FreeAnchor：学习锚点匹配的视觉目标检测方法。载于*Advances in Neural Information Processing Systems*，页码147–155，2019年。2 [40] 朱晨晨、何一辉和Marios Savvides。特征选择性无锚模块的单次目标检测。载于*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*，页码840–849，2019年。8 [41] 朱曦洲、胡涵、林斯蒂芬和戴继峰。可变形卷积网络v2：更灵活，更优结果。载于*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*，页码9308–9316，2019年。8 [42] 朱曦洲、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。*arXiv preprint arXiv:2010.04159*，2020年。2, 6