PIX2SEQ: A LANGUAGE MODELING FRAMEWORK FOR OBJECT DETECTION

Ting Chen, Saurabh Saxena, Lala Li, David J. Fleet, Geoffrey Hinton
Google Research, Brain Team

ABSTRACT

We present *Pix2Seq*, a simple and generic framework for object detection. Unlike existing approaches that explicitly integrate prior knowledge about the task, we cast object detection as a language modeling task conditioned on the observed pixel inputs. Object descriptions (e.g., bounding boxes and class labels) are expressed as sequences of discrete tokens, and we train a neural network to perceive the image and generate the desired sequence. Our approach is based mainly on the intuition that if a neural network knows about where and what the objects are, we just need to teach it how to read them out. Beyond the use of task-specific data augmentations, our approach makes minimal assumptions about the task, yet it achieves competitive results on the challenging COCO dataset, compared to highly specialized and well optimized detection algorithms.[1]

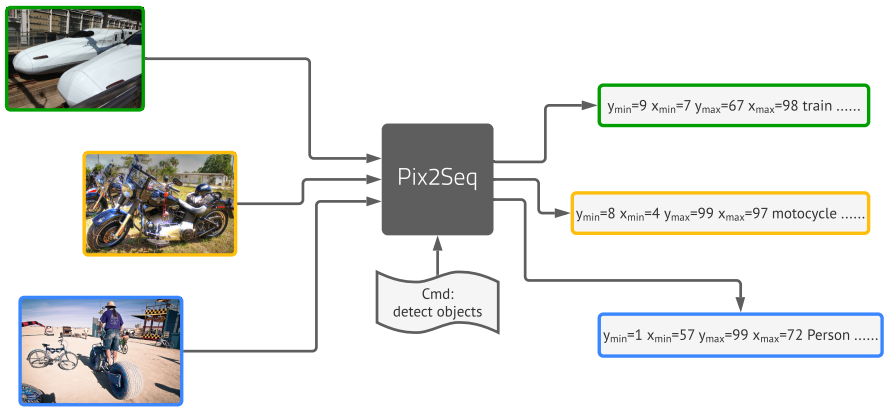


Figure 1: Illustration of Pix2Seq framework for object detection. The neural net perceives an image and generates a sequence of tokens that correspond to bounding boxes and class labels.

1 INTRODUCTION

Visual object detection systems aim to recognize and localize all objects of pre-defined categories in an image. The detected objects are typically described by a set of bounding boxes and associated class labels. Given the difficulty of the task, most existing methods, such as (Girshick, 2015; Ren et al., 2015; He et al., 2017; Lin et al., 2017b; Carion et al., 2020), are carefully designed and highly customized, with a significant amount of prior knowledge in the choice of architecture and loss function. For example, many architectures are tailored to the use of bounding boxes (e.g., with region proposals (Girshick, 2015; Ren et al., 2015) and RoI pooling (Girshick et al., 2014; He et al., 2017)). Others are tied to the use of object queries for object binding (Carion et al., 2020). Loss functions are often similarly tailored to the use of bounding boxes, such as box regression (Szegedy et al., 2013; Lin et al., 2017b), set-based matching (Erhan et al., 2014; Carion et al., 2020), or by incorporating

Correspondence to: iamtingchen@google.com
[1]Code and checkpoints available at https://github.com/google-research/pix2seq.

PIX2SEQ：一种面向目标检测的语言建模框架

陈挺、索拉布·萨克塞纳、拉拉·李、大卫·J·弗利特、杰弗里·辛顿　　谷歌研究院，Brain团队

摘要

我们提出了*Pix2Seq*，一个简单而通用的目标检测框架。与现有方法不同，后者显式地整合了任务相关的先验知识，而我们将目标检测视为一种基于观察到的像素输入的条件语言建模任务。目标描述（如边界框和类别标签）被表达为离散标记的序列，我们训练神经网络来感知图像并生成所需的序列。我们的方法主要基于这样一种直觉：如果一个神经网络已经知道目标的位置和内容，我们只需教会它如何将这些信息读取出来。除了使用任务特定的数据增强外，我们的方法对任务本身的假设极少，但在具有挑战性的COCO数据集上，与高度专业化且经过充分优化的检测算法相比，仍取得了具有竞争力的结果。[1]

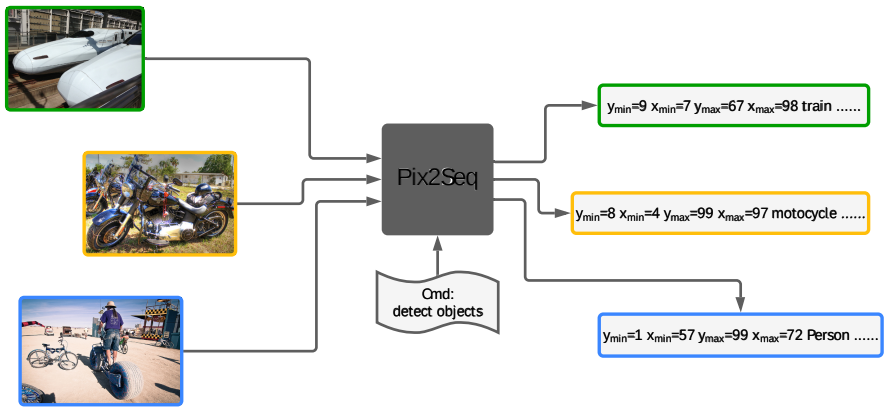


图1：Pix2Seq框架在目标检测中的示意图。神经网络感知图像并生成一系列对应于边界框和类别标签的标记。

1 引言

视觉目标检测系统旨在识别并定位图像中所有预定义类别的物体。检测到的物体通常由一组边界框及对应的类别标签描述。鉴于该任务的复杂性，现有大多数方法（如Girshick, 2015; Ren et al., 2015; He et al., 2017; Lin et al., 2017b; Carion et al., 2020）都经过精心设计且高度定制化，在架构选择和损失函数中融入了大量先验知识。例如，许多架构专门针对边界框的使用进行优化（如采用区域提议（Girshick, 2015; Ren et al., 2015）和RoI池化（Girshick et al., 2014; He et al., 2017））。另一些方法则与用于物体绑定的对象查询机制紧密关联（Carion et al., 2020）。损失函数同样常针对边界框的使用进行专门设计，例如框回归（Szegedy et al., 2013; Lin et al., 2017b）、基于集合的匹配（Erhan et al., 2014; Carion et al., 2020），或通过整合{v*}

Correspondence to: iamtingchen@google.com

[1]Code and checkpoints available at https://github.com/google-research/pix2seq.

specific performance metrics, like intersection-over-union on bounding boxes (Rezatofighi et al., 2019). Although existing systems find applications in myriad domains, from self-driving cars (Sun et al., 2020), to medical image analysis (Jaeger et al., 2020), to agriculture (Sa et al., 2016), the specialization and complexity make them difficult to integrate into a larger system, or generalize to a much broader array of tasks associated with general intelligence.

This paper advocates a new approach, based on the intuition that if a neural net knows about where and what the objects are, we just need to teach it to read them out. And by learning to "describe" objects the model can learn to ground the "language" on pixel observations, leading to useful object representations. This is realized with our Pix2Seq framework (see Figure 1). Given an image, our model produces a sequence of discrete tokens that correspond to object descriptions (e.g., object bounding boxes and class labels), reminiscent of an image captioning system (Vinyals et al., 2015b; Karpathy & Fei-Fei, 2015; Xu et al., 2015). In essence, we cast object detection as a language modeling task conditioned on pixel inputs, for which the model architecture and loss function are generic and relatively simple, without being engineered specifically for the detection task. As such, one can readily extend the framework to different domains or applications, or incorporate it into a perceptual system supporting general intelligence, for which it provides a language interface to a wide range of vision tasks.

To tackle the detection task with Pix2Seq, we first propose a quantization and serialization scheme that converts bounding boxes and class labels into sequences of discrete tokens. We then leverage an encoder-decoder architecture for perceiving pixel inputs and generating the target sequence. The objective function is simply the maximum likelihood of tokens conditioned on pixel inputs and the preceding tokens. While both the architecture and loss function are task-agnostic (without assuming prior knowledge about object detection, e.g., bounding boxes), we can still incorporate task-specific prior knowledge with a sequence augmentation technique, proposed below, that alters both input and target sequences during training. Through extensive experimentation, we demonstrate that this simple Pix2Seq framework can achieve competitive results on the COCO dataset compared to highly customized, well established approaches, including Faster R-CNN (Ren et al., 2015) and DETR (Carion et al., 2020). By pretraining our model on a larger object detection dataset, its performance can be further improved.

2 The Pix2Seq framework

In the proposed Pix2Seq framework we cast object detection as a language modeling task, conditioned on pixel inputs (Figure 1). The system consists of four main components (Figure 2):

- *Image Augmentation*: As is common in training computer vision models, we use image augmentations to enrich a fixed set of training examples (e.g., with random scaling and crops).
- *Sequence construction & augmentation*: As object annotations for an image are usually represented as a *set* of bounding boxes and class labels, we convert them into a *sequence* of discrete tokens.
- *Architecture*: We use an encoder-decoder model, where the encoder perceives pixel inputs, and the decoder generates the target sequence (one token at a time).
- *Objective/loss function*: The model is trained to maximize the log likelihood of tokens conditioned on the image and the preceding tokens (with a softmax cross-entropy loss).

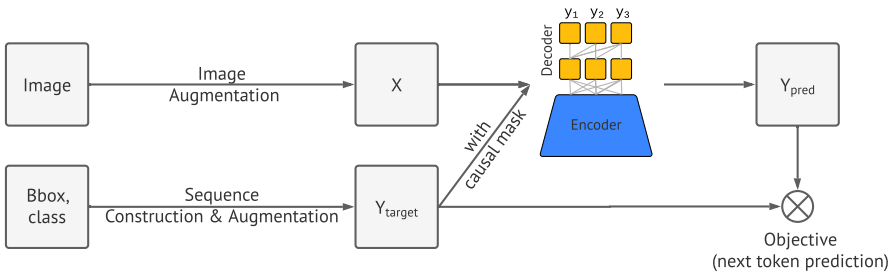


Figure 2: Major components of the Pix2Seq learning framework.

特定性能指标，如边界框的交并比（Rezatofighi等人，2019年）。尽管现有系统已在众多领域找到应用，从自动驾驶汽车（Sun等人，2020年）到医学图像分析（Jaeger等人，2020年），再到农业（Sa等人，2016年），但其专业性和复杂性使得它们难以集成到更大的系统中，或泛化至与通用智能相关的更广泛任务阵列。

本文提出了一种新方法，其核心思想是：若神经网络已掌握物体位置与类别的识别能力，我们只需教会它如何输出这些信息。通过让模型学习"描述"物体，它能将"语言"与像素观察结果建立关联，从而形成有效的物体表征。我们通过Pix2Seq框架实现了这一构想（见图1）。给定一张图像，我们的模型会生成与物体描述（如物体边界框和类别标签）对应的离散标记序列，其原理类似于图像描述生成系统（Vinyals等人，2015b；Karpathy & Fei-Fei，2015；Xu等人，2015）。本质上，我们将物体检测任务转化为基于像素输入的语言建模任务，其模型架构与损失函数具有通用性且相对简单，无需专门针对检测任务进行设计。因此，该框架可轻松扩展至不同领域或应用场景，或整合到支持通用智能的感知系统中——通过语言接口为各类视觉任务提供服务。

为了利用Pix2Seq解决检测任务，我们首先提出了一种量化和序列化方案，将边界框和类别标签转化为离散标记序列。随后，我们采用编码器-解码器架构来感知像素输入并生成目标序列。目标函数简化为在像素输入及前序标记条件下的标记最大似然。尽管该架构与损失函数均与任务无关（无需预设目标检测的先验知识，如边界框），我们仍可通过下文提出的序列增强技术融入任务特定先验知识——该技术会在训练时同步修改输入与目标序列。大量实验表明，这一简洁的Pix2Seq框架在COCO数据集上能取得与高度定制化、成熟方法（如Faster R-CNN（Ren等人，2015）和DETR（Carion等人，2020））相竞争的结果。通过在更大规模的目标检测数据集上预训练模型，其性能还可进一步提升。

2 PIX2SEQ框架

在提出的Pix2Seq框架中，我们将目标检测任务转化为基于像素输入的语言建模任务（图1）。该系统包含四个主要组成部分（图2）：

- *Image Augmentation*在训练计算机视觉模型时，通常会采用图像增强技术来丰富固定的训练样本集（例如，通过随机缩放和裁剪等方式）。
- *Sequence construction & augmentation*由于图像中的对象标注通常以边界框和类别标签的*set*形式表示，我们将其转换为离散标记的*sequence*。
- *Architecture*我们采用编码器-解码器模型，其中编码器感知像素输入，解码器则逐步生成目标序列（每次一个标记）。
- *Objective/loss function*该模型训练的目标是最大化在给定图像及先前标记条件下标记的对数似然（采用softmax交叉熵损失函数）。

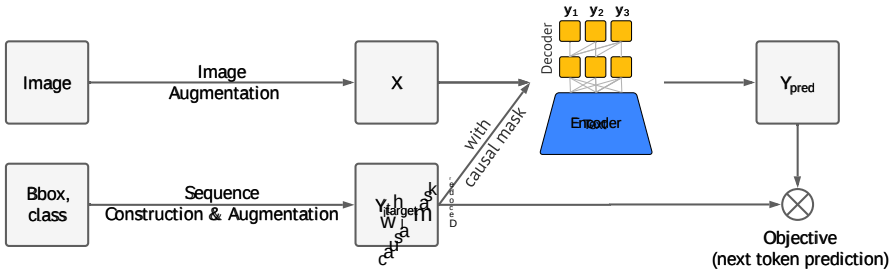


图2：Pix2Seq学习框架的主要组成部分。

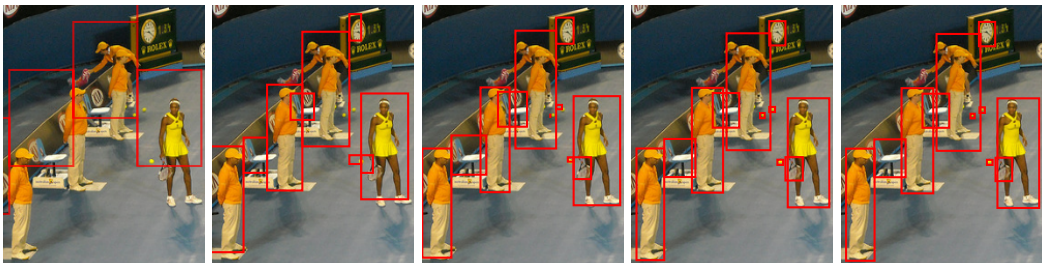
2.1 SEQUENCE CONSTRUCTION FROM OBJECT DESCRIPTIONS

In common object detection datasets, such as Pascal VOC (Everingham et al., 2010), COCO (Lin et al., 2014), and OpenImages (Kuznetsova et al., 2020), images have variable numbers of objects, represented as sets of bounding boxes and class labels. In Pix2Seq we express them as sequences of discrete tokens.

While class labels are naturally expressed as discrete tokens, bounding boxes are not. A bounding box is determined by two of its corner points (i.e., top-left and bottom-right), or by its center point plus height and width. We propose to discretize the continuous numbers used to specify the x, y coordinates of corner points (similarly for height and width if the other box format is used). Specifically, an object is represented as a sequence of five discrete tokens, i.e. $[y_{\text{min}}, x_{\text{min}}, y_{\text{max}}, x_{\text{max}}, c]$, where each of the continuous corner coordinates is uniformly discretized into an integer between $[1, n_{\text{bins}}]$, and c is the class index. We use a shared vocabulary for all tokens, so the vocabulary size is equal to number of bins + number of classes. This quantization scheme for the bounding boxes allows us to use a small vocabulary while achieving high precision. For example, a 600×600 image requires only 600 bins to achieve zero quantization error. This is much smaller than modern language models with vocabulary sizes of 32K or higher (Radford et al., 2018; Devlin et al., 2018). The effect of different levels of quantization on the placement of bounding boxes is illustrated in Figure 3.

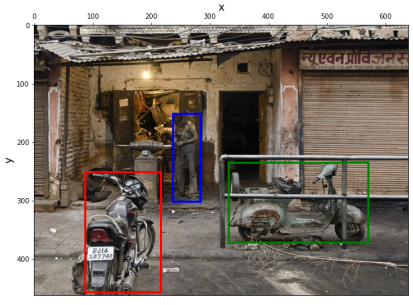
With each object description expressed as a short discrete sequence, we next need to serialize multiple object descriptions to form a single sequence for a given image. Since order of objects does not matter for the detection task per se, we use a random ordering strategy (randomizing the order objects each time an image is shown). We also explore other deterministic ordering strategies, but we hypothesize that random ordering will work just as well as any deterministic ordering, given a capable neural net and autoregressive modeling (where the net can learn to model the distribution of remaining objects conditioned on those observed).

Finally, because different images often have different numbers of objects, the generated sequences will have different lengths. To indicate the end of a sequence, we therefore incorporate an EOS token. The sequence construction process with different ordering strategies is illustrated in Figure 4.



(a) $n_{\text{bins}} = 10$ (b) $n_{\text{bins}} = 50$ (c) $n_{\text{bins}} = 100$ (d) $n_{\text{bins}} = 500$ (e) Original

Figure 3: Applying the proposed discritization of bounding box on an image of 480×640. Only a quarter of the image is shown for better clarity. With a small number of bins, such as 500 bins (∼1 pixel/bin), it achieves high precision even for small objects.



Random ordering (multiple samples):

327 370 653 444 1001 544 135 987 338 1004 508 518 805 892 1004 0

544 135 987 338 1004 327 370 653 444 1001 508 518 805 892 1004 0

508 518 805 892 1004 544 135 987 338 1004 327 370 653 444 1001 0

Area ordering:

544 135 987 338 1004 508 518 805 892 1004 327 370 653 444 1001 0

Dist2ori ordering:

544 135 987 338 1004 327 370 653 444 1001 508 518 805 892 1004 0

Figure 4: Examples of sequence construction with $n_{\text{bins}} = 1000$, and 0 is EOS token.

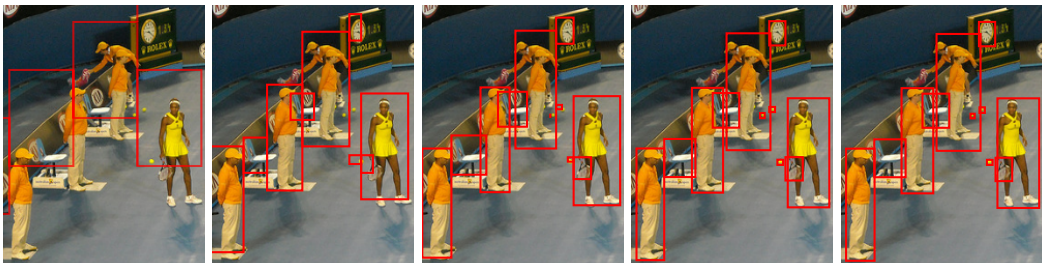
2.1 从对象描述构建序列

在常见的物体检测数据集中，如Pascal VOC（Everingham等人，2010）、COCO（Lin等人，2014）和OpenImages（Kuznetsova等人，2020），图像包含数量不定的物体，以边界框和类别标签的集合形式表示。在Pix2Seq中，我们将它们表达为离散标记的序列。

虽然类别标签自然以离散标记的形式表达，但边界框则不然。边界框由其两个角点（即左上角和右下角）确定，或通过中心点加高度和宽度来定义。我们提出将用于指定角点x, y坐标的连续数值离散化（若采用另一种边界框格式，则对高度和宽度同样处理）。具体而言，一个对象被表示为五个离散标记的序列，即$[y_{\text{min}}, x_{\text{min}}, y_{\text{max}}, x_{\text{max}}, c]$，其中每个连续的角点坐标被均匀离散化为$[1, n_{\text{bins}}]$之间的整数，而$c$为类别索引。所有标记共享同一词汇表，因此词汇表大小等于分箱数+加上类别数。这种边界框的量化方案使我们能够使用较小的词汇表，同时实现高精度。例如，一幅600×600的图像仅需600个分箱即可实现零量化误差。这远小于现代语言模型通常32K或更大的词汇量（Radford等人，2018；Devlin等人，2018）。图3展示了不同量化级别对边界框定位的影响。

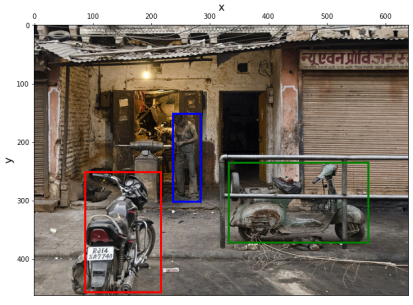
将每个物体描述表达为简短的离散序列后，我们接下来需要将多个物体描述序列化，以形成给定图像的单一序列。由于物体顺序对检测任务本身并不重要，我们采用了随机排序策略（每次展示图像时随机打乱物体顺序）。我们还探索了其他确定性排序策略，但我们假设，只要具备强大的神经网络和自回归建模能力（网络能够学习根据已观察到的物体来建模剩余物体的分布），随机排序的效果将与任何确定性排序策略同样出色。

最后，由于不同图像中的物体数量往往不同，生成的序列也会有不同的长度。因此，我们引入了一个EOS标记来表示序列的结束。图4展示了采用不同排序策略时的序列构建过程。



(a) $n_{\text{bins}} = 10$ (b) $n_{\text{bins}} = 50$ (c) $n_{\text{bins}} = 100$ (d) $n_{\text{bins}} = 500$ (e) Original

图3：在480×640的图像上应用所提出的边界框离散化方法。为更清晰展示，仅显示图像的四分之一区域。即使采用较少的区间数量，例如500个区间（~1像素/区间），该方法对小物体也能实现高精度检测。



Random ordering (multiple samples):

327 370 653 444 1001 544 135 987 338 1004 508 518 805 892 1004 0

544 135 987 338 1004 327 370 653 444 1001 508 518 805 892 1004 0

508 518 805 892 1004 544 135 987 338 1004 327 370 653 444 1001 0

Area ordering:

544 135 987 338 1004 508 518 805 892 1004 327 370 653 444 1001 0

Dist2ori ordering:

544 135 987 338 1004 327 370 653 444 1001 508 518 805 892 1004 0

图4：使用$n_{\text{bins}} = 1000$构建序列的示例，0为EOS标记。

2.2 ARCHITECTURE, OBJECTIVE AND INFERENCE

Treating the sequences that we construct from object descriptions as a "dialect", we turn to generic architectures and objective functions that have been effective in language modeling.

Architecture We use an encoder-decoder architecture. The encoder can be a general image encoder that perceives pixels and encodes them into hidden representations, such as a ConvNet (LeCun et al., 1989; Krizhevsky et al., 2012; He et al., 2016), Transformer (Vaswani et al., 2017; Dosovitskiy et al., 2020), or their combination (Carion et al., 2020). For generation we use a Transformer decoder, widely used in modern language modeling (Radford et al., 2018; Raffel et al., 2019). It generates one token at a time, conditioned on the preceding tokens and the encoded image representation. This removes the complexity and customization in architectures of modern object detectors, e.g., bounding box proposal and regression, since tokens are generated from a single vocabulary with a softmax.

Objective Similar to language modeling, Pix2Seq is trained to predict tokens, given an image and preceding tokens, with a maximum likelihood loss, i.e.,

$$\text{maximize} \sum_{j=1}^{L} \boldsymbol{w}_j \log P(\tilde{\boldsymbol{y}}_j | \boldsymbol{x}, \boldsymbol{y}_{1:j-1}) \ , \tag{1}$$

where $\boldsymbol{x}$ is a given image, $\boldsymbol{y}$ and $\tilde{\boldsymbol{y}}$ are input and target sequences associated with $\boldsymbol{x}$, and L is the target sequence length. $\boldsymbol{y}$ and $\tilde{\boldsymbol{y}}$ are identical in the standard language modeling setup, but they can also be different (as in our later augmented sequence construction). Also, $\boldsymbol{w}_j$ is a pre-assigned weight for j-th token in the sequence. We set $\boldsymbol{w}_j = 1, \forall j$, however it would be possible to weight tokens by their types (e.g., coordinate vs class tokens), or by the size of the corresponding object.

Inference At inference time, we sample tokens from model likelihood, i.e., $P(\boldsymbol{y}_j | \boldsymbol{x}, \boldsymbol{y}_{1:j-1})$. This can be done by either taking the token with the largest likelihood ($\arg\max$ sampling), or using other stochastic sampling techniques. We find that using nucleus sampling (Holtzman et al., 2019) leads to higher recall than $\arg\max$ sampling (Appendix C). The sequence ends when the EOS token is generated. Once the sequence is generated, it is straight-forward to extract and de-quantize the object descriptions (i.e., obtaining the predicted bounding boxes and class labels).

2.3 SEQUENCE AUGMENTATION TO INTEGRATE TASK PRIORS

The EOS token allows the model to decide when to terminate generation, but in practice we find that the model tends to finish without predicting all objects. This is likely due to 1) annotation noise (e.g., where annotators did not identify all the objects), and 2) uncertainty in recognizing or localizing some objects. While this only affects the overall performance by a small percentage (e.g., 1-2% in average precision), it has a larger effect on recall. To encourage higher recall rates, one trick is to delay the sampling of the EOS token by artificially decreasing its likelihood. However, this often leads to noisy and duplicated predictions. In part, this difficult trade-off between precision and recall is a consequence of our model being task agnostic, unaware of the detection task per se.

To mitigate the problem we simply introduce a sequence augmentation technique, thereby incorporating prior knowledge about the task. The target sequence $\tilde{\boldsymbol{y}}$ in conventional autoregressive language modeling (i.e., with no sequence augmentation) is the same as the input sequence $\boldsymbol{y}$. And all tokens in a sequence are real (e.g., converted from human annotations). With sequence augmentation, we instead augment input sequences during training to include both real and synthetic noise tokens. We also modify target sequences so that the model can learn to identify the noise tokens rather than mimic them. This improves the robustness of the model against noisy and duplicated predictions (particularly when the EOS token is delayed to increase recall). The modifications introduced by sequence augmentation are illustrated in Figure 5, and detailed below.

Altered sequence construction We first create *synthetic noise objects* to augment input sequences in the following two ways: 1) adding noise to existing ground-truth objects (e.g., random scaling or shifting their bounding boxes), and 2) generating completely random boxes (with randomly associated class labels). It is worth noting that some of these noise objects may be identical to, or overlapping with, some of the ground-truth objects, simulating noisy and duplicated predictions, as demonstrated

2.2 架构、目标与推理

将我们从物体描述中构建的序列视为一种“方言”，我们转而采用在语言建模中已被证明有效的通用架构和目标函数。

架构 我们采用编码器-解码器架构。编码器可以是通用的图像编码器，负责感知像素并将其编码为隐藏表示，例如ConvNet（LeCun等人，1989；Krizhevsky等人，2012；He等人，2016）、Transformer（Vaswani等人，2017；Dosovitskiy等人，2020）或二者的结合（Carion等人，2020）。在生成部分，我们使用Transformer解码器，该结构在现代语言建模中广泛应用（Radford等人，2018；Raffel等人，2019）。它每次生成一个标记，基于先前生成的标记和编码后的图像表示进行条件生成。由于所有标记均通过单一词汇表的softmax生成，这消除了现代目标检测器架构中的复杂性和定制化需求，例如边界框提议和回归过程。

目标与语言建模类似，Pix2Seq的训练目标是在给定图像及先前标记的条件下预测标记，采用最大似然损失函数，即

$$\text{maximize} \sum_{j=1}^{L} \boldsymbol{w}_j \log P(\tilde{\boldsymbol{y}}_j | \boldsymbol{x}, \boldsymbol{y}_{1:j-1}) \,, \tag{1}$$

其中$\boldsymbol{x}$是给定的图像，$\boldsymbol{y}$和$\tilde{\boldsymbol{y}}$是与$\boldsymbol{x}$相关联的输入和目标序列，L是目标序列的长度。在标准的语言建模设置中，$\boldsymbol{y}$和$\tilde{\boldsymbol{y}}$是相同的，但它们也可以不同（如我们后续增强序列构建中的情况）。此外，$\boldsymbol{w}_j$是序列中第j个令牌的预设权重。我们设定$\boldsymbol{w}_j = 1, \forall j$，但也可以根据令牌类型（如坐标令牌与类别令牌）或对应对象的大小来加权。

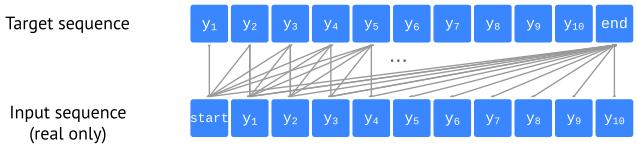
在推理阶段，我们从模型似然中采样标记，即$P(\boldsymbol{y}_j | \boldsymbol{x}, \boldsymbol{y}_{1:j-1})$。这可以通过选择具有最大似然的标记（arg max采样）实现，或采用其他随机采样技术。我们发现，使用核心采样（Holtzman等人，2019年）相比arg max采样能带来更高的召回率（附录C）。当生成EOS标记时，序列终止。序列生成后，直接提取并反量化对象描述（即获得预测的边界框和类别标签）即可。

2.3 序列增强以整合任务先验

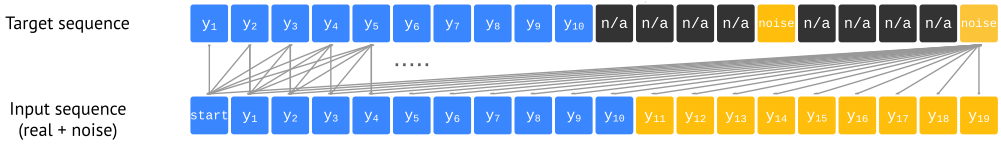
EOS令牌允许模型决定何时终止生成，但在实践中我们发现，模型倾向于在不预测所有对象的情况下结束。这可能是由于：1）标注噪声（例如，标注者未识别出所有对象），以及2）某些对象识别或定位的不确定性。虽然这对整体性能的影响较小（例如，平均精度下降1-2%），但对召回率的影响更为显著。为提高召回率，一种技巧是通过人为降低EOS令牌的采样概率来延迟其采样。然而，这往往会导致预测结果出现噪声和重复。部分而言，这种精确度与召回率之间的艰难权衡，源于我们的模型对任务本身无感知，不了解检测任务的具体要求。

为解决这一问题，我们简单地引入了一种序列增强技术，从而融入了关于该任务的先验知识。在传统的自回归语言建模中（即未采用序列增强时），目标序列$\tilde{\boldsymbol{y}}$与输入序列$\boldsymbol{y}$相同，且序列中的所有标记均为真实标记（例如，由人工标注转换而来）。而通过序列增强，我们在训练过程中对输入序列进行扩充，使其同时包含真实标记和合成的噪声标记。同时，我们调整目标序列，使模型学会识别而非模仿这些噪声标记。这增强了模型对噪声和重复预测的鲁棒性（尤其是在延迟EOS标记以提高召回率的情况下）。图5展示了序列增强所引入的修改，具体细节如下。

序列构造的修改 我们首先创建$\{\boldsymbol{y}, \tilde{\boldsymbol{y}}\}$以以下两种方式增强输入序列：1) 对现有真实标注对象添加噪声（例如，随机缩放或平移其边界框），2) 生成完全随机的框（附带随机关联的类别标签）。值得注意的是，部分噪声对象可能与某些真实标注对象完全相同或存在重叠，以此模拟噪声和重复预测的情况，如示例所示



(a) Conventional autoregressive langage modeling



(b) Langage modeling with sequence augmentation (e.g. adding noise tokens)

Figure 5: Illustration of language modeling with / without sequence augmentation. With sequence augmentation, input tokens are constructed to include both real objects (blue) and synthetic noise objects (orange). For the noise objects, the model is trained to identify them as the "noise" class, and we set the loss weight of "n/a" tokens (corresponding to coordinates of noise objects) to zero since we do not want the model to mimic them.

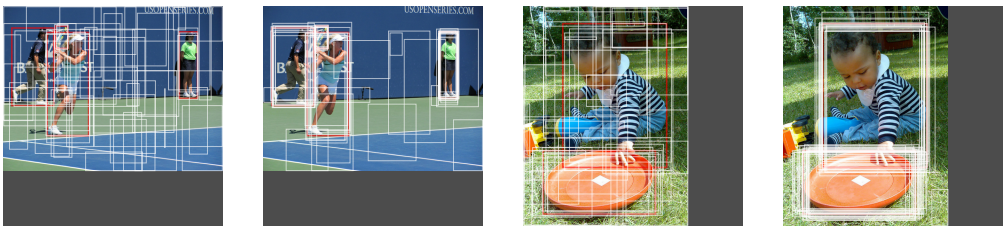


Figure 6: Illustrations of randomly sampled noise objects (in white), vs. ground-truth objects (in red).

in Figure 6. After noise objects are synthesised and discretized, we then append them in the end of the original input sequence. As for the target sequence, we set the target tokens of noise objects to "noise" class (not belonging to any of the ground-truth class labels), and the coordinate tokens of noise objects to "n/a", whose loss weights are set to zero, i.e., setting $\boldsymbol{w}_j = \mathbb{1}_{[\tilde{\boldsymbol{y}}_j \neq \text{“n/a”}]}$ in Eq 1.

Altered inference With sequence augmentation, we are able to substantially delay the EOS token, improving recall without increasing the frequency of noisy and duplicated predictions. Thus, we let the model predict to a maximum length, yielding a fixed-sized list of objects. When we extract the list of bounding boxes and class labels from the generated sequences, we replace the "noise" class label with a real class label that has the highest likelihood among all real class labels. We use the likelihood of the selected class token as a (ranking) score for the object.

3 Experiments

3.1 Experimental Setup

We evaluate the proposed method on the MS-COCO 2017 detection dataset (Lin et al., 2014), containing 118k training images and 5k validation images. To compare with DETR and Faster R-CNN, we report average precision (AP), an integral metric over multiple thresholds, on validation set at the last training epoch. We employ two training strategies: 1) *training from scratch* on COCO in order to compare fairly with the baselines, and also 2) *pretraining+finetuning*, i.e., pretrain the Pix2Seq model on a larger object detection dataset, namely Objects365 (Shao et al., 2019), and then finetune the model on COCO. Since our approach incorporates zero inductive bias / prior knowledge of the object detection task, we expect the second training strategy to be superior.

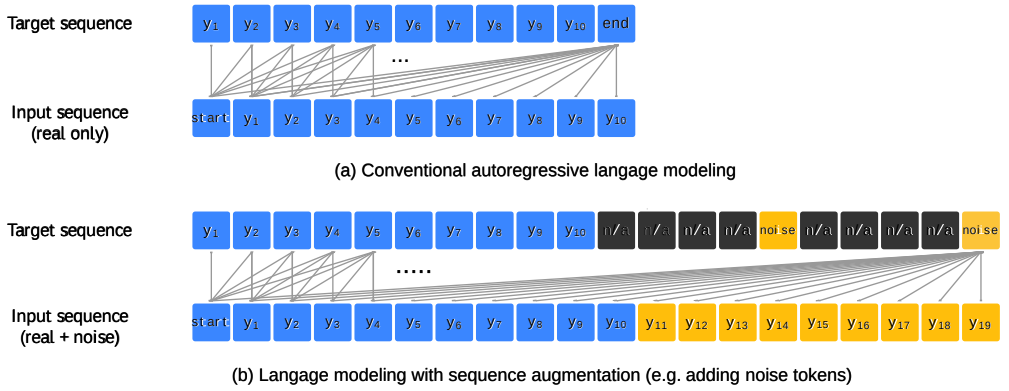


图5：序列增强与无序列增强下的语言建模示意图。在序列增强的情况下，输入令牌被构建为包含真实对象（蓝色）和合成噪声对象（橙色）。对于噪声对象，模型被训练将其识别为“噪声”类，并且我们将“n/a”令牌（对应噪声对象的坐标）的损失权重设为零，因为我们不希望模型模仿它们。

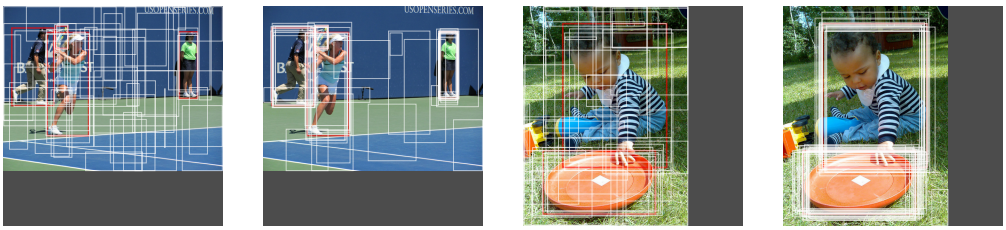


图6：随机采样的噪声对象（白色）与真实对象（红色）的对比示意图。

在图6中。合成并离散化噪声对象后，我们将其附加到原始输入序列的末尾。对于目标序列，我们将噪声对象的目标标记设为“噪声”类（不属于任何真实类别标签），并将噪声对象的坐标标记设为“n/a”，其损失权重设为零，即在公式1中设置$\boldsymbol{w}_j = 1_{[\tilde{\boldsymbol{y}}_j \neq \text{“n/a”}]}$。

通过序列增强，我们能够显著延迟EOS（结束符）标记的出现，在不增加噪声和重复预测频率的情况下提升召回率。因此，我们让模型预测至最大长度，生成一个固定大小的对象列表。当从生成的序列中提取边界框和类别标签列表时，我们会将“噪声”类别标签替换为所有真实类别标签中具有最高似然值的实际类别标签，并将所选类别标记的似然值作为该对象的（排序）得分。

3 实验

3.1 实验设置

我们在MS-COCO 2017检测数据集（Lin等人，2014）上评估所提出的方法，该数据集包含118k训练图像和5k验证图像。为了与DETR和Faster R-CNN进行比较，我们在最后一个训练周期报告验证集上的平均精度（AP），这是一个跨多阈值的综合指标。我们采用两种训练策略：1）*training from scratch*在COCO上公平比较基线方法，以及2）*pretraining+finetuning*，即在更大的目标检测数据集Objects365（Shao等人，2019）上预训练Pix2Seq模型，然后在COCO上微调模型。由于我们的方法融入了零归纳偏差/目标检测任务的先验知识，我们预期第二种训练策略表现更优。

Table 1: Comparison of average precision, over multiple thresholds and object sizes, on COCO validation set. Each section compares different methods of the similar ResNet "backbone". Our models achieve competitive results to both Faster R-CNN and DETR baselines.

Method	Backbone	#params	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster R-CNN	R50-FPN	42M	40.2	61.0	43.8	24.2	43.5	52.0
Faster R-CNN+	R50-FPN	42M	42.0	62.1	45.5	26.6	45.4	53.4
DETR	R50	41M	42.0	62.4	44.2	20.5	45.8	61.1
Pix2seq (Ours)	R50	37M	**43.0**	61.0	45.6	25.1	46.9	59.4
Faster R-CNN	R101-FPN	60M	42.0	62.5	45.9	25.2	45.6	54.6
Faster R-CNN+	R101-FPN	60M	44.0	63.9	47.8	27.2	48.1	56.0
DETR	R101	60M	43.5	63.8	46.4	21.9	48.0	61.8
Pix2seq (Ours)	R101	56M	**44.5**	62.8	47.5	26.0	48.2	60.3
Faster R-CNN	R50-DC5	166M	39.0	60.5	42.3	21.4	43.5	52.5
Faster R-CNN+	R50-DC5	166M	41.1	61.4	44.3	22.9	45.9	55.0
DETR	R50-DC5	41M	**43.3**	63.1	45.9	22.5	47.3	61.1
Pix2seq (Ours)	R50-DC5	38M	**43.2**	61.0	46.1	26.6	47.0	58.6
DETR	R101-DC5	60M	**44.9**	64.7	47.7	23.7	49.5	62.3
Pix2seq (Ours)	R101-DC5	57M	**45.0**	63.2	48.6	28.2	48.9	60.4

For training from scratch, we follow (Carion et al., 2020) using a ResNet backbone (He et al., 2016), followed by 6 layers of transformer encoder and 6 layers of (causal) transformer decoder (Vaswani et al., 2017). We resize images (with a fixed aspect ratio) so the longer side is 1333 pixels. For sequence construction, we use 2000 quantization bins, and we randomize the order of objects every time an image is shown. We append noise objects to real objects such that each image contains 100 objects in total, and hence a sequence length of 500. The model is trained for 300 epochs with a batch size of 128.

For pretraining on Objects365 dataset, we use similar settings as above with a few differences. Notably, instead of using the large 1333×1333 image size, we use a smaller image size of 640×640, and pretrain the models for 400K steps with batch size of 256. It is worth noting that this pretraining process is even faster than training from scratch due to the use of smaller image size. During the fine-tuning on COCO dataset, only a small number of epochs (e.g., 20 to 60 epochs) are needed to achieve good results. And we could use larger image size during fine-tuning as well. Due to the use of larger pretraining dataset, we also experiment with larger models with Vision Transformers (Dosovitskiy et al., 2020).

More details for both training strategies can be found in Appendix B. As for ablations, we use a ResNet-101 backbone with a smaller image size (the longer side is 640), and we train the model from scratch for 200 epochs.

3.2 Main comparisons

Training from scratch on COCO We mainly compare with two widely recognized baselines: DETR and Faster R-CNN. DETR and our model have comparable architectures, but our Transformer decoder does not require learned "object queries" or separated heads for box regression and classification, since our model generates different types of tokens (e.g., coordinate and class tokens) with a single softmax. Faster R-CNN is a well established method, with optimized architectures such as feature-pyramid networks (FPN) (Lin et al., 2017a). Faster R-CNN is typically trained in fewer epochs than DETR or our model, likely because it explicitly incorporates prior knowledge of the task in the architecture itself. Thus we also include an improved Faster R-CNN baseline, denoted as Faster R-CNN+, from (Carion et al., 2020), where Faster R-CNN models are trained with the GIoU loss (Rezatofighi et al., 2019), train-time random crop augmentations, and the long $9\times$ training schedule.

Results are shown in Table 1, where each section compares different methods of the same ResNet "backbone". Overall, Pix2Seq achieves competitive results to both baselines. Our model performs comparably to Faster R-CNN on small and medium objects, but better on larger objects. Compared

表1：在COCO验证集上，针对多阈值及不同物体尺寸的平均精度比较。各部分对比了采用相似ResNet“骨干”网络的不同方法。我们的模型在Faster R-CNN和DETR基线模型上均取得了具有竞争力的结果。

Method	Backbone	#params	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster R-CNN	R50-FPN	42M	40.2	61.0	43.8	24.2	43.5	52.0
Faster R-CNN+	R50-FPN	42M	42.0	62.1	45.5	26.6	45.4	53.4
DETR	R50	41M	42.0	62.4	44.2	20.5	45.8	61.1
Pix2seq (Ours)	R50	37M	**43.0**	61.0	45.6	25.1	46.9	59.4
Faster R-CNN	R101-FPN	60M	42.0	62.5	45.9	25.2	45.6	54.6
Faster R-CNN+	R101-FPN	60M	44.0	63.9	47.8	27.2	48.1	56.0
DETR	R101	60M	43.5	63.8	46.4	21.9	48.0	61.8
Pix2seq (Ours)	R101	56M	**44.5**	62.8	47.5	26.0	48.2	60.3
Faster R-CNN	R50-DC5	166M	39.0	60.5	42.3	21.4	43.5	52.5
Faster R-CNN+	R50-DC5	166M	41.1	61.4	44.3	22.9	45.9	55.0
DETR	R50-DC5	41M	**43.3**	63.1	45.9	22.5	47.3	61.1
Pix2seq (Ours)	R50-DC5	38M	**43.2**	61.0	46.1	26.6	47.0	58.6
DETR	R101-DC5	60M	**44.9**	64.7	47.7	23.7	49.5	62.3
Pix2seq (Ours)	R101-DC5	57M	**45.0**	63.2	48.6	28.2	48.9	60.4

在从头开始训练时，我们遵循（Carion等人，2020）的方法，采用ResNet主干网络（He等人，2016），后接6层Transformer编码器和6层（因果）Transformer解码器（Vaswani等人，2017）。我们将图像（保持固定宽高比）调整尺寸，使较长边为1333像素。在序列构建中，我们使用2000个量化区间，并在每次图像展示时随机化对象顺序。我们将噪声对象附加到真实对象上，使每张图像总共包含100个对象，因此序列长度为500。模型训练共进行300个周期，批次大小为128。

在Objects365数据集上进行预训练时，我们采用了与上述相似的设置，但存在几点差异。值得注意的是，我们没有使用较大的1333×1333图像尺寸，而是采用了较小的640×640尺寸，并以256的批量大小对模型进行了400K步的预训练。需要指出的是，由于使用了较小的图像尺寸，这一预训练过程甚至比从头开始训练还要快。在对COCO数据集进行微调时，仅需少量周期（例如20至60个周期）即可获得良好效果。此外，在微调阶段我们也可以使用更大的图像尺寸。得益于更大规模的预训练数据集，我们还尝试了基于Vision Transformers（Dosovitskiy等人，2020）的更大模型。

两种训练策略的更多细节可在附录B中找到。至于消融实验，我们采用ResNet-101主干网络，并缩小图像尺寸（较长边为640），从头开始训练模型200个周期。

3.2 主要比较

在COCO上从头训练 我们主要与两个广泛认可的基线方法进行比较DETR和Faster R-CNN。DETR与我们的模型架构相似，但我们的Transformer解码器无需学习“对象查询”或为边界框回归与分类设置独立头部，因为我们的模型通过单一softmax生成不同类型的令牌（如坐标令牌和类别令牌）。Faster R-CNN是一种成熟的方法，采用了如特征金字塔网络（FPN）（Lin等人，2017a）等优化架构。Faster R-CNN通常比DETR或我们的模型训练周期更短，这可能是因为其架构本身显式融入了任务先验知识。因此，我们还引入了一个改进的Faster R-CNN基线，记为Faster R-CNN{v*}（源自Carion等人2020年的工作），其中Faster R-CNN模型采用GIoU损失（Rezatofighi等人，2019）、训练时随机裁剪增强以及长达9倍的训练计划进行训练。

结果如表1所示，其中各部分比较了同一ResNet“主干”网络的不同方法。总体而言，Pix2Seq取得了与两种基线方法相当的结果。我们的模型在中小型物体上表现与Faster R-CNN相当，但在较大物体上表现更优。相比

Table 2: Average precision of finetuned Pix2seq models on COCO with different backbone architectures and image sizes. All models are pretrained on Objects365 dataset. As a comparison, our best model without pretraining obtains 45.0 AP (in Table 1) with image size of 1333×1333. The pretraining is with 640×640 image size while fine-tuning (a few epochs) can use larger image sizes.

Backbone	# params	Image size during finetuning		
		640×640	1024×1024	1333×1333
R50	37M	39.1	41.7	42.6
R50-C4	85M	44.7	46.9	47.3
ViT-B	115M	44.2	46.5	47.1
ViT-L	341M	47.6	49.0	50.0

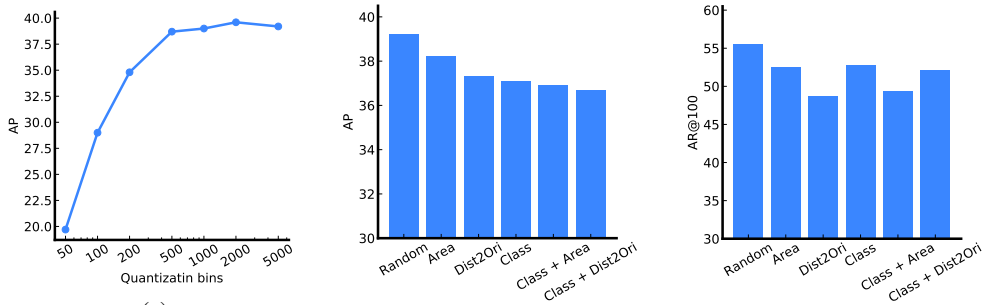
with DETR, our model performs comparably or slightly worse on large and medium objects, but substantially better (4-5 AP) on small objects.

Pretrain on Objects365 and finetune on COCO As shown in Table 2, the performances of Objects365 pretrained Pix2Seq models are strong across various model sizes and image sizes. The best performance (with 1333 image size) is 50 AP which is 5% higher than the best model trained from scratch, and the performance holds up very well even with 640 image size. Notably, with a smaller image size used for pretraining, the pretrain+finetune process is faster than training from scratch, and also generalizes better. Both factors are crucial for training larger and better models.

3.3 Ablation on sequence construction

Figure 7a explores the effect of coordinate quantization on performance. For this ablation we consider images the longest size of which is 640 pixels. The plot indicates that quantization to 500 bins or more is sufficient; with 500 bins there are approximately 1.3 pixels per bin, which does not introduce significant approximation error. Indeed, as long as one has as many bins as the number of pixels (along the longest side of the image) there should be no significant error due to quantization of the bounding box coordinates.

We also consider different object ordering strategies in sequence construction during training. These include 1) random, 2) area (i.e., descending object size), 3) dist2ori (i.e., the distance of top-left corner of the bounding box to the origin), 4) class (name), 5) class + area (i.e., the objects are first ordered by their class, and if there are multiple objects of the same class, they are ordered by area), and 6) class + dist2ori. Figure 7b shows average precision (AP) and Figure 7c shows average recall (AR) at the top-100 predictions. Both in terms of precision and recall, the random ordering yields the best performance. We conjecture that with deterministic ordering, it may be difficult for the model to recover from mistakes of missing objects made earlier on, while with random ordering it would still be possible to retrieve them later.



(a) (b) (c)

Figure 7: Ablations on sequence construction. (a) Quantization bins vs. performance. (b) and (c) show AP and AR@100 for different object ordering strategies.

表2：不同主干架构和图像尺寸下，微调后的Pix2seq模型在COCO上的平均精度。所有模型均在Objects365数据集上进行了预训练。作为对比，我们未经预训练的最佳模型在1333×1333的图像尺寸下取得了45.0 AP（见表1）。预训练采用640×640的图像尺寸，而微调（少量周期）可使用更大的图像尺寸。

Backbone	# params	Image size during finetuning		
		640×640	1024×1024	1333×1333
R50	37M	39.1	41.7	42.6
R50-C4	85M	44.7	46.9	47.3
ViT-B	115M	44.2	46.5	47.1
ViT-L	341M	47.6	49.0	50.0

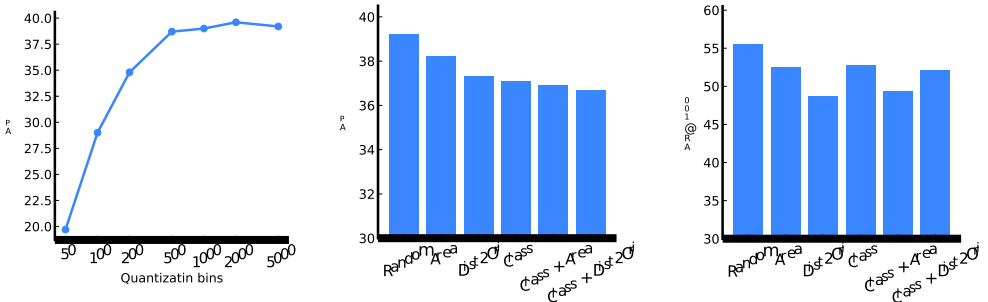
使用DETR时，我们的模型在大型和中型物体上的表现相当或略逊一筹，但在小型物体上表现显著更优（提升4-5 AP）。

在Objects365上预训练并在COCO上微调 如表2所示，经过Objects365预训练的Pix2Seq模型在不同模型规模和图像尺寸下均表现出色。最佳性能（1333图像尺寸）达到50 AP，比从头训练的顶级模型高出5%，且即便采用640图像尺寸，性能仍保持优异。值得注意的是，当预训练使用较小图像尺寸时，预训练+微调过程比从头训练更快，且泛化能力更强。这两点对于训练更大更优模型至关重要。

3.3 序列构建的消融研究

图7a探讨了坐标量化对性能的影响。在此消融实验中，我们考虑最长边为640像素的图像。图表显示，量化至500个或更多区间已足够；500个区间意味着每个区间约对应1.3个像素，这不会引入显著的近似误差。事实上，只要区间数量与图像最长边的像素数量相当，边界框坐标的量化就不会导致显著误差。

在训练过程中，我们还考虑了序列构建时的不同对象排序策略。这些策略包括：1）随机排序，2）按面积排序（即对象大小降序），3）按dist2ori排序（即边界框左上角到原点的距离），4）按类别（名称）排序，5）按类别+面积排序（即首先按类别对对象进行排序，若同一类别有多个对象，则按面积排序），以及6）按类别+dist2ori排序。图7b展示了前100个预测的平均精度（AP），图7c则展示了相应的平均召回率（AR）。无论是精度还是召回率，随机排序均表现出最佳性能。我们推测，确定性排序可能导致模型难以从早期遗漏对象的错误中恢复，而随机排序则仍有机会在后续阶段检索到这些对象。



(a) (b) (c)

图7：序列构建的消融实验。(a) 量化分箱与性能关系。(b)和(c)展示了不同物体排序策略下的平均精度(AP)和召回率@100(AR@100)。

3.4 ABLATION ON SEQUENCE AUGMENTATION

Here we study the impact of sequence augmentation (i.e., adding the noise objects) for both model training strategies: 1) training from scratch on COCO, and 2) pretraining on Objects365 and finetuning on COCO. Results for training from scratch w/wo sequence augmentation are shown in Figure 8, and we find that without sequence augmentation, the AP is marginally worse if one delays the sampling of EOS token during the inference (via likelihood offsetting), but the recall is significantly worse for the optimal AP. Table 3 shows similar results for pretraining+finetuning setting (where we set a loss weight of 0.1 on ending token instead of tuning their likelihood offset), and we find that AP is not significantly affected while recall is significantly worse without sequence augmentation. It is also worth noting that sequence augmentation is mainly effective during the fine-tuning.

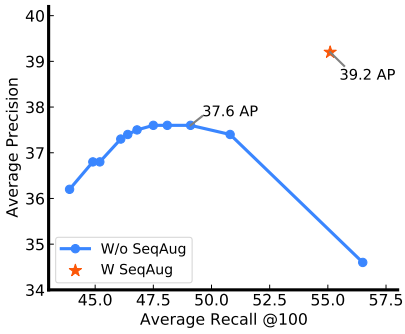


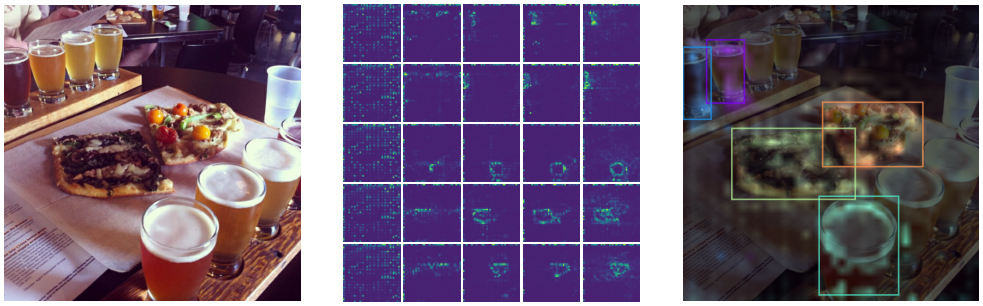
Figure 8: Impact of sequence augmentation on when training from scratch on COCO.

SeqAug in Pretrain	SeqAug in Finetune	AP	AR@100
✗	✗	43.7	55.4
✗	✓	44.5	61.6
✓	✓	44.7	61.7

Table 3: Impact of sequence augmentation when pretraining on Objects365 and finetuning on COCO. Sequence augmentation has a major impact on average recall (@100) but a smaller influence on AP. Most improvements can be achieved during fine-tuning.

3.5 VISUALIZATION OF DECODER'S CROSS ATTENTION MAP

When generating a new token, the transformer decoder uses self attention over the preceding tokens and cross attention over the encoded visual feature map. Here we visualize the cross attention (averaged over layers and heads) as the model predicts a new token. Figure 9 shows cross attention maps as the first few tokens are generated. One can see that the attention is very diverse when predicting the first coordinate token (i.e y_{min}), but then quickly concentrates and fixates on the object.



(a) (b) (c)

Figure 9: Decoder's cross attention to visual feature map when predicting the first 5 objects. (b) we reshape a prediction sequence of 25 into a 5x5 grid, so each row represents a prediction for 5 tokens $[y_{\text{min}}, x_{\text{min}}, y_{\text{max}}, x_{\text{max}}, c]$. The attention is diverse when selecting the first token of the object, then quickly concentrates on the object. (c) Overlay of the cross attention (when predicting the class token) on the original image.

3.4 序列增强的消融研究

在此，我们研究了序列增强（即添加噪声对象）对两种模型训练策略的影响：1) 在COCO上从头开始训练，2) 在Objects365上预训练并在COCO上微调。图8展示了从头训练时使用与不使用序列增强的结果，我们发现，若不采用序列增强，在推理过程中延迟EOS令牌采样（通过似然偏移）时AP值会略微下降，但对于最优AP而言召回率则显著降低。表3显示了预训练+微调设置下的类似结果（此处我们对结束令牌设置了0.1的损失权重而非调整其似然偏移），可见序列增强缺失时AP未受显著影响，但召回率明显下降。值得注意的是，序列增强主要在微调阶段发挥显著作用。

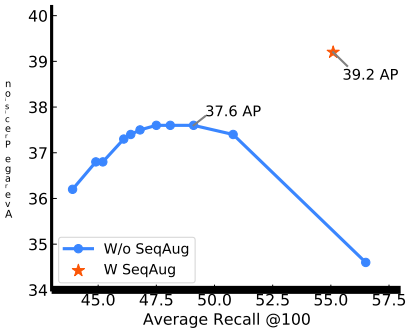


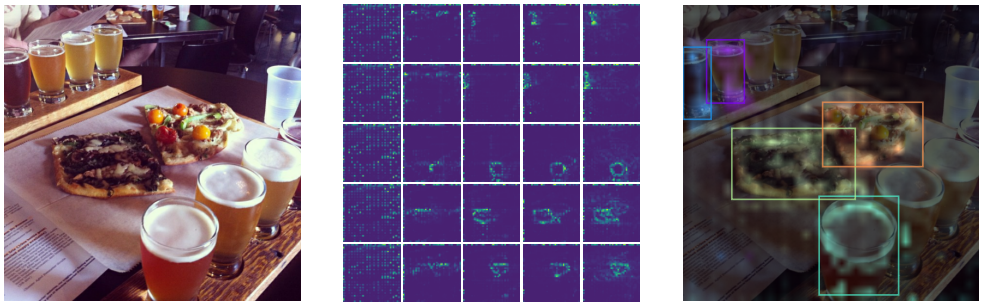
图8：在COCO上从头训练时序列增强的影响。

SeqAug in Pretrain	SeqAug in Finetune	AP	AR@100
✗	✗	43.7	55.4
✗	✓	44.5	61.6
✓	✓	44.7	61.7

表3：在Objects365上预训练并在COCO上微调时序列增强的影响。序列增强对平均召回率（@100）有显著提升，但对AP的影响较小。大部分改进可在微调阶段实现。

3.5 解码器交叉注意力图的可视化

在生成新标记时，Transformer解码器会对先前的标记进行自注意力计算，并对编码后的视觉特征图进行交叉注意力计算。这里我们将模型预测新标记时的交叉注意力（各层与注意力头的平均值）可视化。图9展示了生成最初几个标记时的交叉注意力分布图。可以看出，在预测第一个坐标标记（即y_{min}）时，注意力分布非常分散，但随后迅速集中并锁定在目标物体上。



(a) (b) (c)

图9：解码器在预测前5个物体时对视觉特征图的交叉注意力。(b) 我们将25个预测序列重塑为5x5网格，因此每一行代表对5个标记$[y_{\text{min}}, x_{\text{min}}, y_{\text{max}}, x_{\text{max}}, c]$的预测。在选择物体首个标记时注意力分布较为分散，随后迅速集中到该物体上。(c) 原始图像上叠加了（预测类别标记时的）交叉注意力分布。

4 RELATED WORK

Object detection. Existing object detection algorithms incorporate explicit prior knowledge about the task in their choice of architecture and loss function. To predict a set of bounding boxes, architectures of modern detectors are specifically designed to produce a large set of proposals (Girshick, 2015; Ren et al., 2015; Cai & Vasconcelos, 2018), anchors (Lin et al., 2017b), or window centers (Tian et al., 2019; Zhou et al., 2019). Non-maximum suppression (Bodla et al., 2017) is often required to prevent duplicate predictions. While DETR (Carion et al., 2020) avoids sophisticated bounding box proposals and non-maximum suppression, it still requires a set of learned "object queries", specially for object binding. These detectors all require sub-networks (or extra layers) separately for regressing bounding boxes and class labels. Pix2Seq avoids such complexities by having a generic image encoder and sequence decoder, with a single softmax for producing coordinate tokens and class labels.

Beyond architectures, the loss functions of existing detectors are also highly tailored for matching bounding boxes. For example, the loss function is often based on bounding box regression (Szegedy et al., 2013; Lin et al., 2017b), intersection over union (Rezatofighi et al., 2019), and set-based matching (Erhan et al., 2014; Liu et al., 2016; Redmon et al., 2016; Stewart et al., 2016; Carion et al., 2020). Pix2Seq avoids specialized losses, showing that a straightforward maximum likelihood objective with softmax cross entropy can work well.

Our work is also related to recurrent models in object detection (Stewart et al., 2016; Park & Berg, 2015; Romera-Paredes & Torr, 2016; Salvador et al., 2017; Ren & Zemel, 2017), in which the system learns to predict one object at a time. As above, both architecture and loss functions in these approaches are often tailored to the detection task. Furthermore, these approaches are not based on Transformers, and have not been evaluated against modern baselines on larger datasets.

Language modeling. Our work is inspired by recent success of modern language modeling (Radford et al., 2019; Raffel et al., 2019; Brown et al., 2020). Although originally intended for natural languages, the underlying methodology has been shown capable of modeling various sequential data, such as machine translation (Sutskever et al., 2014; Bahdanau et al., 2014), image captioning (Vinyals et al., 2015b; Karpathy & Fei-Fei, 2015; Xu et al., 2015), and many others (Vinyals et al., 2015a; Huang et al., 2018; Ramesh et al., 2021; Chen et al., 2021). Our work enriches this portfolio and shows that it works for even non-sequential data (by turning a set of objects into a sequence of tokens). We augment both input and target sequences for our model to incorporate task-specific prior knowledge; similar sequence corruption scheme have been used in language models (Devlin et al., 2018; Clark et al., 2020), and bear some similarity to noise-contrastive learning (Gutmann & Hyvärinen, 2010) and the discriminator in GANs (Goodfellow et al., 2014).

5 CONCLUSION AND FUTURE WORK

This paper introduces Pix2Seq, a simple yet generic framework for object detection. By casting object detection as a language modeling task, our approach largely simplifies the detection pipeline, removing most of the specialization in modern detection algorithms. We believe that our framework not only works for object detection, but can also be applied to other vision tasks where the output can be represented by a relatively concise sequence of discrete tokens (e.g., keypoint detection, image captioning, visual question answering). To this end, we hope to extend Pix2Seq as a generic and unified interface for solving a large variety of vision tasks.

A major limitation of our approach is that autoregressive modeling is expensive for long sequences (mainly during model inference). Practical measures to mitigate the issue includes: 1) stop inference when the ending token is produced (e.g., in COCO dataset, there are, in average, 7 objects per image, leading to a relatively small number of $\sim$35 tokens), 2) applying it to offline inference, or online scenarios where the objects of interest are relatively sparse (e.g. locate a specific object with language description). However, future work is needed to make it faster for real-time object detection applications. Another limitation is that the current approach for training Pix2Seq is entirely based on human annotation, and by reducing such dependence, it can enable the model to benefit from more unlabeled data.

4 相关工作

目标检测。现有的目标检测算法在其架构选择和损失函数设计中融入了关于任务的显式先验知识。为了预测一组边界框，现代检测器的架构专门设计用于生成大量提议（Girshick, 2015; Ren et al., 2015; Cai & Vasconcelos, 2018）、锚框（Lin et al., 2017b）或窗口中心（Tian et al., 2019; Zhou et al., 2019）。通常需要非极大值抑制（Bodla et al., 2017）来防止重复预测。尽管DETR（Carion et al., 2020）避免了复杂的边界框提议和非极大值抑制，但它仍需要一组学习到的“对象查询”，专门用于对象绑定。这些检测器均需单独的子网络（或额外层）来回归边界框和类别标签。Pix2Seq通过采用通用的图像编码器和序列解码器，并仅使用单一softmax生成坐标标记和类别标签，从而避免了这些复杂性。

除了架构之外，现有检测器的损失函数也高度针对边界框匹配进行了定制。例如，损失函数通常基于边界框回归（Szegedy等人，2013；Lin等人，2017b）、交并比（Rezatofighi等人，2019）以及基于集合的匹配（Erhan等人，2014；Liu等人，2016；Redmon等人，2016；Stewart等人，2016；Carion等人，2020）。Pix2Seq避免了专门的损失函数，表明采用带softmax交叉熵的简单最大似然目标也能取得良好效果。

我们的工作还与目标检测中的循环模型相关（Stewart等人，2016；Park & Berg，2015；Romera-Paredes & Torr，2016；Salvador等人，2017；Ren & Zemel，2017），这些模型通过学习一次预测一个目标来实现检测。如上所述，这些方法中的架构和损失函数通常针对检测任务进行了专门设计。此外，这些方法并非基于Transformer架构，也未在更大规模的数据集上与现代基准进行过对比评估。

语言建模。我们的工作受到现代语言建模近期成功的启发（Radford等人，2019；Raffel等人，2019；Brown等人，2020）。尽管最初是为自然语言设计的，但该方法论已被证明能够建模多种序列数据，如机器翻译（Sutskever等人，2014；Bahdanau等人，2014）、图像描述生成（Vinyals等人，2015b；Karpathy & Fei-Fei，2015；Xu等人，2015）以及许多其他领域（Vinyals等人，2015a；Huang等人，2018；Ramesh等人，2021；Chen等人，2021）。我们的研究丰富了这一系列成果，并展示了该方法甚至适用于非序列数据（通过将一组对象转化为标记序列）。我们为模型增强了输入和目标序列，以融入任务特定的先验知识；类似的序列扰动策略已在语言模型中使用（Devlin等人，2018；Clark等人，2020），并与噪声对比学习（Gutmann & Hyvärinen，2010）及生成对抗网络中的判别器（Goodfellow等人，2014）存在一定相似性。

5 结论与未来工作

本文介绍了Pix2Seq，一个简单而通用的目标检测框架。通过将目标检测任务转化为语言建模问题，我们的方法极大地简化了检测流程，消除了现代检测算法中的大部分专业化设计。我们相信，该框架不仅适用于目标检测，还能应用于其他视觉任务——只要其输出可表示为相对简洁的离散标记序列（例如关键点检测、图像描述生成、视觉问答等）。为此，我们希望将Pix2Seq扩展为一个通用且统一的接口，用于解决各类视觉任务。

我们方法的一个主要局限在于，自回归建模对长序列而言计算成本高昂（尤其在模型推理阶段）。缓解这一问题的实际措施包括：1) 在生成结束标记时停止推理（例如，COCO数据集中平均每张图像包含7个目标，导致~35标记数量相对较少）；2) 将其应用于离线推理，或目标对象相对稀疏的在线场景（如通过语言描述定位特定对象）。然而，要实现实时目标检测应用的速度提升，仍需未来工作加以改进。另一局限是目前Pix2Seq的训练完全依赖人工标注，若能降低这种依赖性，模型将能从更多未标注数据中受益。

ACKNOWLEDGEMENTS

We specially thank Xiuye Gu for preparing the Objects365 dataset. We thank Mohammad Norouzi, Simon Kornblith, Tsung-Yi Lin, Allan Jabri, and Kevin Swersky for the helpful discussions.

REFERENCES

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5561–5569, 2017.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

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*, pp. 6154–6162, 2018.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pp. 213–229. Springer, 2020.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *arXiv preprint arXiv:2106.01345*, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pp. 1597–1607. PMLR, 2020a.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in Neural Information Processing Systems*, 33: 22243–22255, 2020b.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*, 2020.

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.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. Scalable object detection using deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2147–2154, 2014.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2918–2928, 2021.

Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, 2015.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014.

致谢

我们特别感谢顾秀业为准备Objects365数据集所做的贡献。同时，我们也感谢Mohammad Norouzi、Simon Kornblith、林忠毅(Tsung-Yi Lin)、Allan Jabri以及Kevin Swersky在有益讨论中提供的帮助。

参考文献 Dzmitry Bahdanau、Kyunghyun Cho和Yoshua Bengio。通过联合学习对齐与翻译的神经机器翻译。*arXiv preprint arXiv:1409.0473*，2014年。 Navaneeth Bodla、Bharat Singh、Rama Chellappa和Larry S Davis。Soft-NMS——用一行代码提升目标检测。载于*Proceedings of the IEEE International Conference on Computer Vision*，第5561–5569页，2017年。 Tom B Brown、Benjamin Mann、Nick Ryder、Melanie Subbiah、Jared Kaplan、Prafulla Dhariwal、Arvind Neelakantan、Pranav Shyam、Girish Sastry、Amanda Askell等。语言模型是小样本学习者。*arXiv preprint arXiv:2005.14165*，2020年。 Zhaowei Cai和Nuno Vasconcelos。Cascade R-CNN：深入高质量目标检测。载于*Proceedingsof the IEEE Conference on Computer Vision and Pattern Recognition*，第6154–6162页，2018年。 Nicolas Carion、Francisco Massa、Gabriel Synnaeve、Nicolas Usunier、Alexander Kirillov和Sergey Zagoruyko。基于Transformer的端到端目标检测。载于*European Conference on Computer Vision*，第213–229页。Springer，2020年。 Lili Chen、Kevin Lu、Aravind Rajeswaran、Kimin Lee、Aditya Grover、Michael Laskin、Pieter Abbeel、Aravind Srinivas和Igor Mordatch。决策Transformer：通过序列建模实现强化学习。*arXiv preprint arXiv:2106.01345*，2021年。 Ting Chen、Simon Kornblith、Mohammad Norouzi和Geoffrey Hinton。视觉表示对比学习的简单框架。载于*International Conference on Machine Learning*，第1597–1607页。PMLR，2020a。 Ting Chen、Simon Kornblith、Kevin Swersky、Mohammad Norouzi和Geoffrey E Hinton。大型自监督模型是强大的半监督学习者。*Advances in Neural Information Processing Systems*，33卷：22243–22255页，2020b。 Kevin Clark、Minh-Thang Luong、Quoc V. Le和Christopher D. Manning。ELECTRA：将文本编码器预训练为判别器而非生成器。载于*ICLR*，2020年。 Jacob Devlin、Ming-Wei Chang、Kenton Lee和Kristina Toutanova。BERT：用于语言理解的深度双向Transformer预训练。*arXiv preprint arXiv:1810.04805*，2018年。 Alexey Dosovitskiy、Lucas Beyer、Alexander Kolesnikov、Dirk Weissenborn、Xiaohua Zhai、Thomas Unterthiner、Mostafa Dehghani、Matthias Minderer、Georg Heigold、Sylvain Gelly等。一幅图像相当于16x16个词：大规模图像识别的Transformer。载于*International Conference on Learning Representations*，2020年。 Dumitru Erhan、Christian Szegedy、Alexander Toshev和Dragomir Anguelov。使用深度神经网络的可扩展目标检测。载于*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*，第2147–2154页，2014年。 M. Everingham、L. Van Gool、C. K. I. Williams、J. Winn和A. Zisserman。PASCAL视觉对象分类（VOC）挑战赛。*International Journal of Computer Vision*，88(2):303–338，2010年6月。 Golnaz Ghiasi、Yin Cui、Aravind Srinivas、Rui Qian、Tsung-Yi Lin、Ekin D Cubuk、Quoc V Le和Barret Zoph。简单复制粘贴是实例分割的强大数据增强方法。载于*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*，第2918–2928页，2021年。 Ross Girshick。Fast R-CNN。载于*Proceedings of the IEEE International Conference on Computer Vision*，第1440–1448页，2015年。 Ross Girshick、Jeff Donahue、Trevor Darrell和Jitendra Malik。用于精确目标检测和语义分割的丰富特征层次结构。载于*Proceedings of the IEEE Conference on Computer Vision and PatternRecognition*，第580–587页，2014年。 Ian Goodfellow、Jean Pouget-Abadie、Mehdi Mirza、Bing Xu、David Warde-Farley、Sherjil Ozair、Aaron Courville和Yoshua Bengio。生成对抗网络。*Advances in Neural Information Processing Systems*，27卷，2014年。

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth International Conference on artificial intelligence and statistics*, pp. 297–304. JMLR Workshop and Conference Proceedings, 2010.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969, 2017.

Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8129–8138, 2020.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

Andrew G Howard. Some improvements on deep convolutional neural network based image classification. *arXiv preprint arXiv:1312.5402*, 2013.

Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M Dai, Matthew D Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer. *arXiv preprint arXiv:1809.04281*, 2018.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European Conference on Computer Vision*, pp. 646–661. Springer, 2016.

Paul F Jaeger, Simon AA Kohl, Sebastian Bickelhaupt, Fabian Isensee, Tristan Anselm Kuder, Heinz-Peter Schlemmer, and Klaus H Maier-Hein. Retina u-net: Embarrassingly simple exploitation of segmentation supervision for medical object detection. In *Machine Learning for Health Workshop*, pp. 171–183. PMLR, 2020.

Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3128–3137, 2015.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25:1097–1105, 2012.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *International Journal of Computer Vision*, 128(7):1956–1981, 2020.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4): 541–551, 1989.

Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2359–2367, 2017.

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*, pp. 740–755. Springer, 2014.

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*, pp. 2117–2125, 2017a.

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*, pp. 2980–2988, 2017b.

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*, pp. 21–37. Springer, 2016.

迈克尔·古特曼与阿波·许韦里宁。噪声对比估计：非归一化统计模型的新估计原理。载于
Proceedings of the thirteenth International Conference on artificial intelligenceand statistics，第297–304页。JMLR研讨会与会议论文集，2010年。

何恺明、张翔宇、任少卿、孙剑。深度残差学习在图像识别中的应用。载于
Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition，第770–778页，2016年。
何恺明、Georgia Gkioxari、Piotr Dollár和Ross Girshick。Mask R-CNN。载于*Proceedings of the IEEE International Conference on Computer Vision*，第2961–2969页，2017年。

Elad Hoffer、Tal Ben-Nun、Itay Hubara、Niv Giladi、Torsten Hoefler与Daniel Soudry合著。通过实例重复增强批次：提升泛化能力的研究。载于*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*，第8129–8138页，2020年。

阿里·霍尔茨曼、简·布伊斯、杜丽、马克斯韦尔·福布斯与崔艺珍。神经文本退化的奇妙案例。
arXiv preprint arXiv:1904.09751，2019年。

安德鲁·G·霍华德。基于深度卷积神经网络的图像分类方法若干改进。*arXiv preprint arXiv:1312.5402*，2013年。 程志安娜·黄、阿什什·瓦斯瓦尼、雅各布·乌兹科雷特、诺姆·沙泽尔、伊恩·西蒙、柯蒂斯·霍桑、安德鲁·M·戴、马修·D·霍夫曼、莫妮卡·丁库莱斯库与道格拉斯·埃克。音乐变换器。*arXiv preprint arXiv:1809.04281*，2018年。 高黄、孙宇、庄子刘、丹尼娅·塞德拉与基利安·Q·温伯格。随机深度深度网络。载于*European Conference on Computer Vision*，第646–661页。斯普林格，2016年。 保罗·F·耶格尔、西蒙·AA·科尔、塞巴斯蒂安·比克尔豪普特、法比安·伊森西、特里斯坦·安塞尔姆·库德尔、海因茨-彼得·施莱默与克劳斯·H·迈尔-海因。视网膜U-Net：医学目标检测中分割监督的极简利用。载于*Machine Learning for Health Workshop*，第171–183页。PMLR，2020年。 安德烈·卡帕西与李飞飞。生成图像描述的深度视觉-语义对齐方法。载于
Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition，第3128–3137页，2015年。 迪德里克·金马与吉米·巴。Adam：一种随机优化方法。*arXiv preprint arXiv:1412.6980*，2014年。 亚历克斯·克里热夫斯基、伊利亚·苏茨克弗与杰弗里·E·辛顿。基于深度卷积神经网络的ImageNet分类。*Advances in Neural Information Processing Systems*，25:1097–1105，2012年。 阿丽娅·塔兹涅佐娃、哈桑·罗姆、尼尔·阿尔德林、贾斯珀·尤林斯、伊万·克拉辛、约尔迪·蓬特-图塞特、沙哈布·卡马利、斯特凡·波波夫、马泰奥·马洛奇、亚历山大·科列斯尼科夫等。Open Images数据集v4。*International Journal of Computer Vision*，128(7):1956–1981，2020年。 扬·勒昆、伯恩哈德·博瑟、约翰·S·登克、唐尼·亨德森、理查德·E·霍华德、韦恩·哈伯德与劳伦斯·D·杰克尔。反向传播算法在手写邮政编码识别中的应用。*Neural computation*，1(4):541–551，1989年。

李毅, 齐浩志, 戴继峰, 纪翔, 和魏亦忱. 全卷积实例感知语义分割. 载于
Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 第2359–2367页, 2017年.

林宗一、Michael Maire、Serge Belongie、James Hays、Pietro Perona、Deva Ramanan、Piotr Dollár与C Lawrence Zitnick。Microsoft COCO：上下文中的常见物体。载于*European Conference on Computer Vision*，第740–755页。Springer出版社，2014年。

林宗仪、Piotr Dollár、Ross Girshick、何恺明、Bharath Hariharan和Serge Belongie。特征金字塔网络用于目标检测。收录于*Proceedings of the IEEE Conference on Computer Vision andpattern recognition*，第2117–2125页，2017a。

林倞毅、Priya Goyal、Ross Girshick、何恺明与Piotr Dollár。密集目标检测中的焦点损失。《
Proceedings of the IEEE International Conference on Computer Vision》，第2980–2988页，2017b年。

Wei Liu、Dragomir Anguelov、Dumitru Erhan、Christian Szegedy、Scott Reed、Cheng-Yang Fu和Alexander C Berg。SSD：单次多框检测器。载于*European Conference on Computer Vision*，第21–37页。Springer出版社，2016年。

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

Eunbyung Park and Alexander C Berg. Learning to decompose for object detection and instance segmentation. *arXiv preprint arXiv:1511.06449*, 2015.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*, 2021.

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*, pp. 779–788, 2016.

Mengye Ren and Richard S Zemel. End-to-end instance segmentation with recurrent attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6656–6664, 2017.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28:91–99, 2015.

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union. June 2019.

Bernardino Romera-Paredes and Philip Hilaire Sean Torr. Recurrent instance segmentation. In *European Conference on Computer Vision*, pp. 312–329. Springer, 2016.

Inkyu Sa, Zongyuan Ge, Feras Dayoub, Ben Upcroft, Tristan Perez, and Chris McCool. Deepfruits: A fruit detection system using deep neural networks. *sensors*, 16(8):1222, 2016.

Amaia Salvador, Miriam Bellver, Victor Campos, Manel Baradad, Ferran Marques, Jordi Torres, and Xavier Giro-i Nieto. Recurrent neural networks for semantic instance segmentation. *arXiv preprint arXiv:1712.00617*, 2017.

Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8430–8439, 2019.

Russell Stewart, Mykhaylo Andriluka, and Andrew Y Ng. End-to-end people detection in crowded scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2325–2333, 2016.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2446–2454, 2020.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pp. 3104–3112, 2014.

Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep neural networks for object detection. *Advances in neural information processing systems*, 26, 2013.

Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9627–9636, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. Grammar as a foreign language. *Advances in Neural Information Processing Systems*, 28:2773–2781, 2015a.

Ilya Loshchilov与Frank Hutter。解耦权重衰减正则化。见于*International Conference on Learning Representations*，2018年。

Eunbyung Park 与 Alexander C Berg。学习分解以实现目标检测与实例分割。*arXiv preprint arXiv:1511.06449*，2015年。

亚历克·拉德福德、卡蒂克·纳拉辛汉、蒂姆·萨利曼斯与伊利亚·苏茨克维。通过生成式预训练提升语言理解能力。2018。

亚历克·拉德福德、杰弗里·吴、雷文·柴尔德、大卫·吕安、达里奥·阿莫迪、伊利亚·苏茨克弗等。语言模型是无监督多任务学习者。*OpenAI blog*, 1(8):9, 2019年。

Colin Raffel、Noam Shazeer、Adam Roberts、Katherine Lee、Sharan Narang、Michael Matena、Yanqi Z hou、Wei Li和Peter J Liu。探索统一文本到文本转换器在迁移学习中的极限。*arXiv preprint arXiv:1910.10683*，2019年。

Aditya Ramesh、Mikhail Pavlov、Gabriel Goh、Scott Gray、Chelsea Voss、Alec Radford、Mark Chen与Ilya Sutskever。零样本文本到图像生成。*arXiv preprint arXiv:2102.12092*，2021年。

约瑟夫·雷德蒙、桑托什·迪瓦拉、罗斯·吉尔希克与阿里·法哈迪。你只需看一次：统一、实时的目标检测。载于*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*，第779–788页，2016年。 任梦野与理查德·泽梅尔。基于循环注意力的端到端实例分割。载于*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*，第6656–6664页，2017年。 任少卿、何恺明、罗斯·吉尔希克与孙剑。Faster R-CNN：利用区域提议网络实现实时目标检测。*Advances in Neural Information Processing Systems*，28卷：91–99页，2015年。 哈米德·雷扎托菲吉、内森·蔡、郭俊英、阿米尔·萨德吉安、伊恩·里德与西尔维奥·萨瓦雷斯。广义交并比。2019年6月。 伯纳迪诺·罗梅拉-帕雷德斯与菲利普·希莱尔·肖恩·托尔。循环实例分割。载于*European Conference on Computer Vision*，第312–329页。斯普林格出版社，2016年。 Inkyu Sa、葛宗元、Feras Dayoub、Ben Upcroft、Tristan Perez与Chris McCool。DeepFruits：基于深度神经网络的水果检测系统。*sensors*，16卷8期：1222页，2016年。 阿玛亚·萨尔瓦多、米里亚姆·贝尔弗、维克多·坎波斯、马内尔·巴拉达德、费兰·马克斯、霍尔迪·托雷斯与泽维尔·吉罗-尼托。用于语义实例分割的循环神经网络。*arXiv preprint arXiv:1712.00617*，2017年。 邵帅、李泽明、张天元、彭超、余刚、张翔宇、李静与孙剑。Objects365：面向目标检测的大规模高质量数据集。载于*Proceedings of the IEEE/CVF international conference on computer vision*，第8430–8439页，2019年。 拉塞·斯图尔特、米哈伊洛·安德里留卡与吴恩达。拥挤场景中的端到端行人检测。载于*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*，第2325–2333页，2016年。

裴孙、亨里克·克雷茨施马尔、薛西斯·多蒂瓦拉、奥雷连·舒瓦德、维贾伊赛·帕特奈克、保罗·崔、詹姆斯·郭、尹舟、柴玉宁、本杰明·凯恩等。《自动驾驶感知的可扩展性：Waymo开放数据集》。载于*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*，第2446–2454页，2020年。

Ilya Sutskever、Oriol Vinyals和Quoc V Le。用神经网络进行序列到序列学习。载于*Advances in Neural Information Processing Systems*，第3104–3112页，2014年。

克里斯蒂安·塞格迪、亚历山大·托舍夫与杜米特鲁·埃尔汉。用于目标检测的深度神经网络。*Advances in neural information processing systems*，26卷，2013年。

支天、沈春华、陈浩与何通。FCOS：全卷积一阶段目标检测。载于*Proceedings of the IEEE/CVF International Conference on Computer Vision*，第9627–9636页，2019年。

阿希什·瓦斯瓦尼、诺姆·沙泽尔、尼基·帕尔马、雅各布·乌兹科雷特、利昂·琼斯、艾丹·N·戈麦斯、卢卡什·凯泽和伊利亚·波洛苏欣。《注意力就是你所需要的一切》。载于*Advances in Neural Information Processing Systems*，第5998–6008页，2017年。

奥里奥尔·维尼亚尔斯、卢卡什·凯泽、特里·库、斯拉夫·彼得罗夫、伊利亚·苏茨克维尔和杰弗里·辛顿。《语法作为一种外语》。*Advances in Neural Information Processing Systems*，28:2773–2781，2015a。

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3156–3164, 2015b.

Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pp. 2048–2057. PMLR, 2015.

Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

Oriol Vinyals、Alexander Toshev、Samy Bengio和Dumitru Erhan。展示与讲述：一种神经图像描述生成器。载于*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*，第3156–3164页，2015b。 Yuxin Wu、Alexander Kirillov、Francisco Massa、Wan-Yen Lo和Ross Girshick。Detectron 2。https://github.com/facebookresearch/detectron2，2019。 Kelvin Xu、Jimmy Ba、Ryan Kiros、Kyunghyun Cho、Aaron Courville、Ruslan Salakhudinov、Rich Zemel和Yoshua Bengio。展示、关注与讲述：基于视觉注意力的神经图像描述生成。载于*InternationalConference on Machine Learning*，第2048–2057页。PMLR，2015。 Xingyi Zhou、Dequan Wang和Philipp Krähenbühl。以点代物。
arXiv preprint arXiv:1904.07850，2019。

A QUANTIZATION AND DEQUANTIZATION OF COORDINATES

Algorithm 1 and 2 illustrate the quantization and dequantization process of (normalized) coordinates.

Algorithm 1 Quantization of (normalized) coordinates

```
def quantize(x, bins=1000):
  # x is a real number between [0, 1]
  # returns an integer between [0, bins-1]
  return int(x * (bins - 1))
```

Algorithm 2 Dequantization of discrete tokens of coordinates

```
def dequantize(x, bins=1000):
  # x is an integer between [0, bins-1]
  # returns a real number between [0, 1]
  return float(x) / (bins - 1)
```

B TRAINING DETAILS

Training from scratch on COCO For baseline architectures, we follow (Carion et al., 2020) using a ResNet backbone (He et al., 2016), followed by 6 layers of transformer encoder and 6 layers of (causal) transformer decoder (Vaswani et al., 2017). The main dimension of transformer is set to 256 with 8 attention heads, and the dimension of the feed-forward network is set to 1024. We use the stochastic depth (Huang et al., 2016) with a rate of 10% to reduce overfitting. Per (Carion et al., 2020), we also experiment with the DC5 variant of ResNet (Li et al., 2017), which increases the resolution of its output feature map by a factor of two.[2]

For image augmentation during training, we perform scale jittering with random crops (Ghiasi et al., 2021; Wu et al., 2019) with strength of $[0.1, 3]$. We resize images (with a fixed aspect ratio) so the longer side is 1333 pixels. Following (Howard, 2013; Chen et al., 2020a;b), we also use color distortion with a strength of 0.5. For sequence construction, we use 2000 quantization bins, and we randomize the order of objects every time an image is shown. We append noise objects to real objects such that each image contains 100 objects in total, and hence a sequence length of 500.

We train the entire network from scratch for 300 epochs with a batch size of 128. For each image in a mini-batch, we perform two independent augmentations, similar to (Hoffer et al., 2020), resulting in a 256 effective batch size, which we find helpful to reduce overfitting. We use AdamW optimizer (Kingma & Ba, 2014; Loshchilov & Hutter, 2018) with a learning rate of 0.003 and weight decay of 0.05. We use a learning rate warmup for 10 epochs and then linearly decay the learning rate over the course of training.

Pretraining on Objects365 We explore a wider range of architecture variants including both hybrid ResNet and transformer models (Carion et al., 2020), as well as pure transformers based on image patches (Dosovitskiy et al., 2020). The details of the architecture can be found in our released code. Since Objects365 dataset is much larger than COCO (1.7M images vs 118K images), we use a weaker image augmentation (scale jittering range of $[0.3, 2]$ for ViT backbones, and $[0.9, 1.2]$ for ResNet backbones) without color distortion. For sequence construction, we use 1000 quantization bins. And we still apply sequence augmentation with sampled noise objects added by default.

We use a smaller image size of 640×640, and pretrain the models for 400K steps with batch size of 256. We do not perform two augmentations per batch as in training from scratch. And we use a smaller learning rate of 0.001 with the same weight decay of 0.05. We use a cosine learning rate decay with a initial warmup of 20K steps.

As for the finetuning on COCO dataset, we use a batch size of 128 for ResNet backbones, and 64 for ViT backbones. Most models are finetuned for 60 epochs with a learning rate of $3e^{-5}$, but even fewer epochs yield similar results. We still use scale jittering with a range of $[0.3, 2]$ for image augmentation.

[2] Adding a dilation to the last ResNet stage and removing the stride from the first convolution of that stage.

坐标的量化与反量化

算法1和2展示了（归一化）坐标的量化与反量化过程。

算法1 （归一化）坐标的量化

```
def quantize(x, bins=1000): # x 是一个介于 [0, 1] 之间的实数 # 返回一个介于 [0, bins-1] 之间的整数 return int(x * (bins - 1))
```

算法2 坐标离散标记的反量化

```
def 反量化(x, 分箱数=1000):  # x 是一个介于 [0, 分箱数-1] 的整数  # 返回一个介于 [0, 1] 的实数  return float(x) / (分箱数 - 1)
```

B 训练细节

在COCO数据集上从头训练　对于基线架构，我们遵循(Carion等人，2020)的方法，采用ResNet主干网络(He等人，2016)，后接6层Transformer编码器和6层(因果)Transformer解码器(Vaswani等人，2017)。Transformer的主要维度设置为256，配备8个注意力头，前馈网络的维度设为1024。我们使用10%比率的随机深度(Huang等人，2016)来减少过拟合。根据(Carion等人，2020)，我们还尝试了ResNet的DC5变体(Li等人，2017)，该变体将其输出特征图的分辨率提高了一倍。[2]

在训练过程中进行图像增强时，我们采用随机裁剪的尺度抖动方法（Ghiasi等人，2021；Wu等人，2019），强度为[0.1, 3]。我们将图像（保持固定宽高比）调整大小，使较长边为1333像素。遵循（Howard，2013；Chen等人，2020a；b）的方法，我们还使用了强度为0.5的色彩失真。对于序列构建，我们使用2000个量化分箱，并在每次图像展示时随机化对象的顺序。我们将噪声对象附加到真实对象上，使每张图像总共包含100个对象，因此序列长度为500。

我们从头开始训练整个网络，共进行300个周期（epoch），每批（batch）大小为128。对于小批次中的每张图像，我们执行两次独立的增强操作，类似于(Hoffer等人，2020)的做法，从而得到256的有效批次大小，这有助于减少过拟合。我们采用AdamW优化器（Kingma & Ba, 2014; Loshchilov & Hutter, 2018），学习率设为0.003，权重衰减为0.05。训练初期使用10个周期的学习率预热（warmup），之后在整个训练过程中线性衰减学习率。

在Objects365上的预训练　我们探索了更广泛的架构变体，包括混合型ResNet与Transformer模型（Carion等人，2020），以及基于图像块的纯Transformer架构（Dosovitskiy等人，2020）。具体架构细节可在我们公开的代码中查阅。由于Objects365数据集规模远超COCO（170万张图像对比11.8万张图像），我们采用了较弱的图像增强策略（ViT骨干网络的尺度抖动范围为[0.3, 2]，ResNet骨干网络则为[0.9, 1.2]），且未使用色彩失真。在序列构建方面，我们采用1000个量化分箱。默认情况下，我们仍会通过添加采样噪声对象来实施序列增强。

我们采用较小的图像尺寸640{v*}640，并以256的批量大小对模型进行了400K步的预训练。与从头开始训练不同，我们未对每批次执行两次增强。同时，我们使用了较小的学习率0.001，并保持相同的权重衰减率0.05。学习率采用余弦衰减策略，初始预热阶段为20K步。

至于在COCO数据集上的微调，我们为ResNet主干网络使用128的批量大小，ViT主干网络则使用64。大多数模型以$3e^{-5}$的学习率微调60个周期，但更少的周期也能得到相似的结果。我们仍采用范围在[0.3,2]的尺度抖动进行图像增强。

[2]向th添加一个扩张　e last ResNet stage and removing the stride from the first convolution 该阶段的

C ABLATION ON INFERENCE ($\arg\max$ VS NUCLEUS SAMPLING)

Nucleus sampling (Holtzman et al., 2019) has been applied to language modeling to reduce duplication and increase diversity in generated samples. Here we study its impact on sampling from our trained model.

Given the distribution $P(\boldsymbol{y}_j|\boldsymbol{x}, \boldsymbol{y}_{1:j-1})$, to apply nucleus sampling, we first define its top-p vocabulary $V^{(p)} \subset V$ as the smallest set such that

$$\sum_{\boldsymbol{y}_j \in V^{(p)}} P(\boldsymbol{y}_j|\boldsymbol{x}, \boldsymbol{y}_{1:j-1}) \geq p. \tag{2}$$

Let $p' = \sum_{\boldsymbol{y}_j \in V^{(p)}} P(\boldsymbol{y}_j|\boldsymbol{x}, \boldsymbol{y}_{1:j-1})$, and we can re-calibrate the conditional likelihood as following for sampling the next token.

$$P'(\boldsymbol{y}_j|\boldsymbol{x}, \boldsymbol{y}_{1:j-1}) = \begin{cases} P(\boldsymbol{y}_j|\boldsymbol{x}, \boldsymbol{y}_{1:i-1})/p' & \text{if } \boldsymbol{y}_j \in V^{(p)} \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

We vary the hyper-parameter p of nucleus sampling used in generating the output sequence (during inference). When $p = 0$, it corresponds to $\arg\max$ sampling, otherwise it samples from a truncated ranked list of tokens that has a cumsum larger or equal to p. In Figure 10, we see that use of nucleus sampling (with $p > 0$) improves object recall and thus also leads to better average precision. There is a relatively flat region of AP between 0.2 and 0.5, and we select p to be 0.4 as our default value for other experiments.

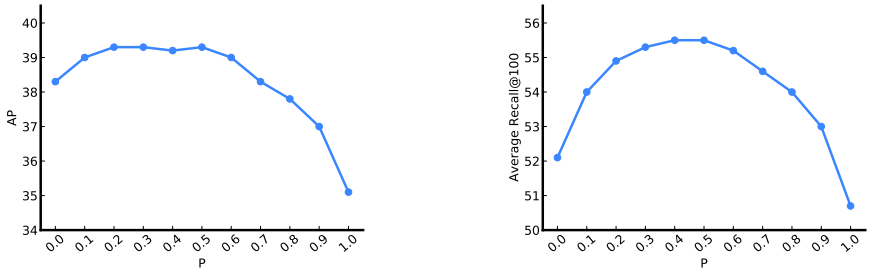


Figure 10: Varying parameter p in nucleus sampling during inference results in different AP and AR. With $p = 0$, it is equivalent to argmax sampling. Sampling with $p > 0$ is helpful for increasing recall (and precision).

D VISUALIZATION OF SIMILARITY AMONG COORDINATE TOKENS

In our model, bounding box coordinates are not represented as floating points, but encoded as discrete tokens. Here we study the similarity among these coordinate tokens via their embeddings. Note that the discrete coordinate tokens and class name tokens are in the same vocabulary and share the same embedding matrix. Specifically, we first slice the learned embedding matrix corresponding to coordinate tokens, and then compute the cosine similarity of embedding vectors for these coordinate tokens.

Figure 11 shows cosine similarity among embeddings of coordinate tokens. We can see that nearby coordinates have higher similarities in their token embeddings than far away ones. This emergent property of our model is likely due to the noises / uncertainties in bounding box annotations (i.e. a bounding box annotation is a random sample from a distribution over potential bounding boxes which encodes locality of coordinates).

E THE ABILITY TO DIRECT THE ATTENTION WITH GIVEN COORDINATES

We explore the model's ability to *pay attention to a pointed region* specified via coordinates. We divide an image evenly into an $N \times N$ grid of rectangular regions, each specified by a sequence of

C 推理消融实验（arg max 对比 核采样）

核采样（Holtzman等人，2019年）已被应用于语言建模中，以减少生成样本的重复性并增加多样性。在此，我们研究了其对从训练模型中采样的影响。

给定分布$P(\boldsymbol{y}_j|\boldsymbol{x}, \boldsymbol{y}_{1:j-1})$，为了应用核采样，我们首先将其top-$p$词汇$V^{(p)} \subset V$定义为满足以下条件的最小集合

$$\sum_{\boldsymbol{y}_j \in V^{(p)}} P(\boldsymbol{y}_j|\boldsymbol{x}, \boldsymbol{y}_{1:j-1}) \geq p. \tag{2}$$

令$p' = \sum_{\boldsymbol{y}_j \in V^{(p)}} P(\boldsymbol{y}_j|\boldsymbol{x}, \boldsymbol{y}_{1:j-1})$，我们可以如下重新校准条件似然以采样下一个标记。

$$P'(\boldsymbol{y}_j|\boldsymbol{x}, \boldsymbol{y}_{1:j-1}) = \begin{cases} P(\boldsymbol{y}_j|\boldsymbol{x}, \boldsymbol{y}_{1:i-1})/p' & \text{if } \boldsymbol{y}_j \in V^{(p)} \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

我们调整了在生成输出序列（推理阶段）时使用的核采样超参数p。当p =为0时，对应的是arg max采样方式；否则，它会从一个累积和大于或等于p的截断排序令牌列表中进行采样。在图10中可以看到，采用核采样（当p >非0时）能够提升物体召回率，从而也带来更高的平均精度。在0.2至0.5之间存在一个相对平坦的平均精度区域，因此我们选择p为0.4作为其他实验的默认值。

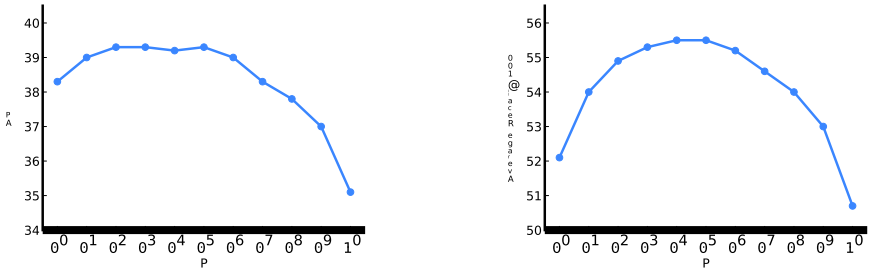


图10：在推理过程中改变参数p进行核采样会导致不同的AP和AR结果。当p =为0时，等同于argmax采样。而采用p >大于0的采样有助于提高召回率（以及精确率）。

坐标令牌间相似性的D可视化

在我们的模型中，边界框坐标并非以浮点数形式表示，而是编码为离散的token。此处我们通过其嵌入向量研究这些坐标token间的相似性。需注意的是，离散坐标token与类别名称token同属一个词汇表，并共享相同的嵌入矩阵。具体而言，我们首先切分出与坐标token对应的学习嵌入矩阵，随后计算这些坐标token嵌入向量间的余弦相似度。

图11展示了坐标标记嵌入之间的余弦相似度。可以看出，邻近坐标在标记嵌入中的相似度高于相距较远的坐标。我们模型表现出的这一特性，很可能源于边界框标注中的噪声/不确定性（即边界框标注是从潜在边界框分布中随机抽取的样本，该分布编码了坐标的局部性）。

E THE ABI 以给定CO引导注意力的能力 纵坐标

我们探究模型通过坐标指定*pay attention to a pointed region*的能力。将图像均匀划分为$N \times N$的矩形区域网格，每个区域由一系列

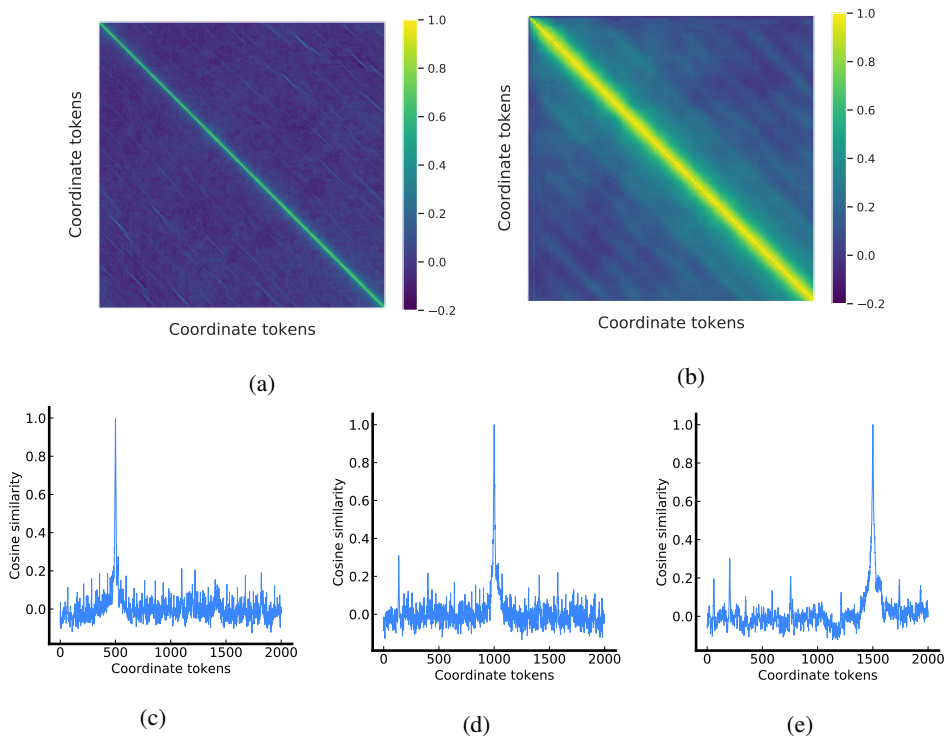
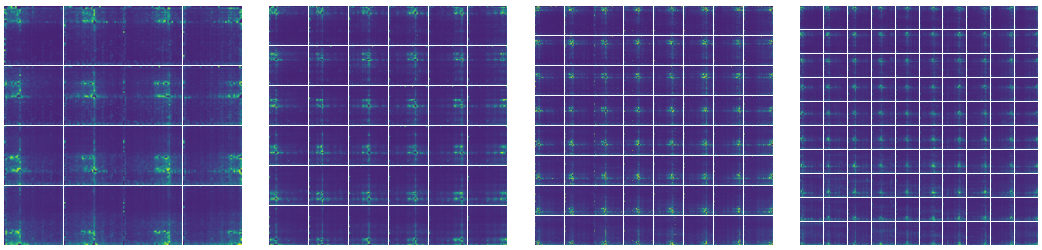


Figure 11: (a) Cosine similarity among embeddings of coordinate tokens. (b) is part of (a) covering only the first 100 tokens. (c), (d) and (e) are the 500-th, 1000-th and 1500-th rows of (a), respectively. Nearby coordinates have higher similarities in their token embeddings.

coordinates for its bounding box. We then visualize the decoder's cross attention to visual feature map after reading the sequence of coordinates for each region, i.e., $[y_{\text{min}}, x_{\text{min}}, y_{\text{max}}, x_{\text{max}}]$. We shuffle the pixels in the image to remove distraction from existing objects, and remove 2% of the top attentions for clarity. Interestingly, as shown in Figure 12, it seems the model can pay attention to the specified region at different scales.

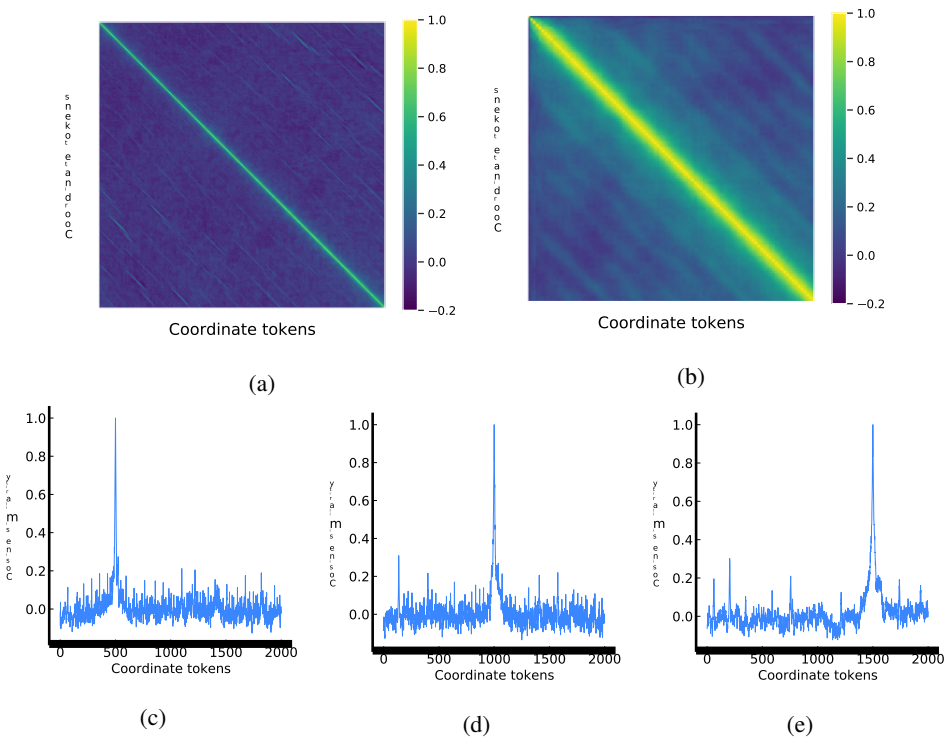


(a) 4×4 grids (b) 6×6 grids (c) 8×8 grids (d) 10×10 grids

Figure 12: Each grid is a visualization of decoder's attention after reading a small sequence of coordinates, i.e., $[y_{\text{min}}, x_{\text{min}}, y_{\text{max}}, x_{\text{max}}]$. Visualization is done for grids of different sizes. The network learns to pay attention to pointed region at different scales.

F More visualization on decoder's cross attention

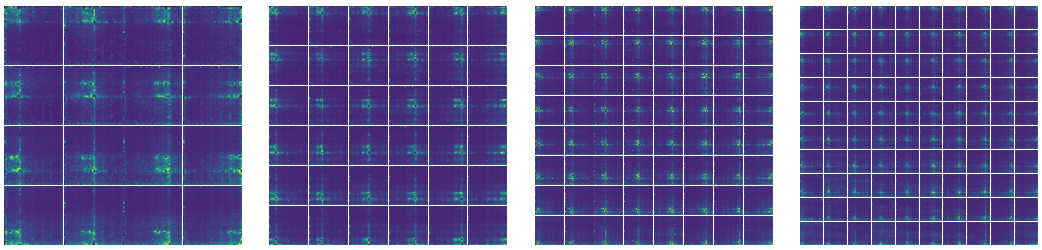
In Figure 13, we overlay the cross attention (when predicting the class token) on the original image for several other images, and it shows that the decoder pays the most attention to the object when predicting the class token.



(a) (b) (c) (d) (e)

图11：(a) 坐标令牌嵌入间的余弦相似度。(b) 是(a)的一部分，仅覆盖前100个令牌。(c)、(d)和(e)分别是(a)的第500、1000和1500行。邻近坐标在其令牌嵌入中具有更高的相似性。

其边界框的坐标。随后，在解码器读取每个区域的坐标序列（即$[y_{\min}, x_{\min}, y_{\max}, x_{\max}]$）后，我们可视化其对视觉特征图的交叉注意力。为了消除现有对象的干扰，我们对图像中的像素进行了随机打乱，并移除了前2%的注意力权重以提高清晰度。有趣的是，如图12所示，模型似乎能够在不同尺度上关注到指定的区域。

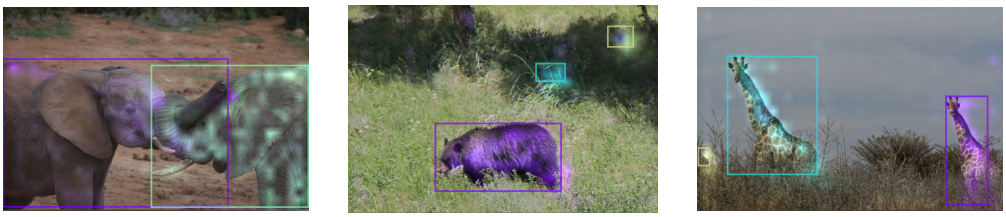


(a) 4×4 grids (b) 6×6 grids (c) 8×8 grids (d) 10×10 grids

图12：每个网格是对解码器在读取一小段坐标序列（即$[y_{\min}, x_{\min}, y_{\max}, x_{\max}]$）后注意力机制的可视化。针对不同尺寸的网格进行了可视化展示。网络学会在不同尺度上关注所指区域。

F 解码器交叉注意力更多可视化

在图13中，我们将交叉注意力（预测类别标记时）叠加到其他几张原始图像上，结果显示解码器在预测类别标记时最为关注目标对象。



(a) (b) (c)

Figure 13: Visualization of Transformer decoder's cross attention (when predicting class tokens) conditioned on the given bounding boxes.

G VISUALIZATION OF DETECTION RESULTS

In Figure 14, we visualize detection results of one of Pix2seq model (with 46 AP) on a subset of images from COCO validation set that contain a crowded set of objects.

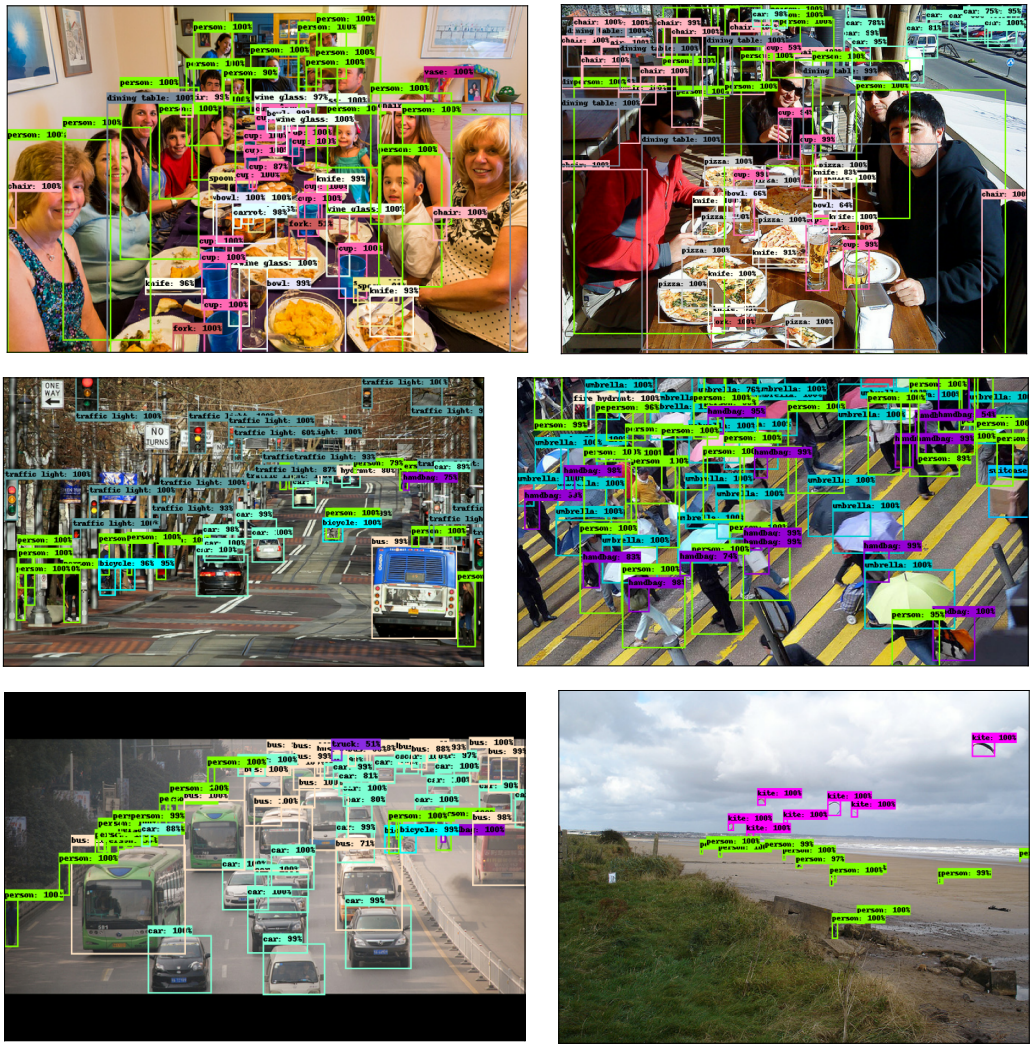
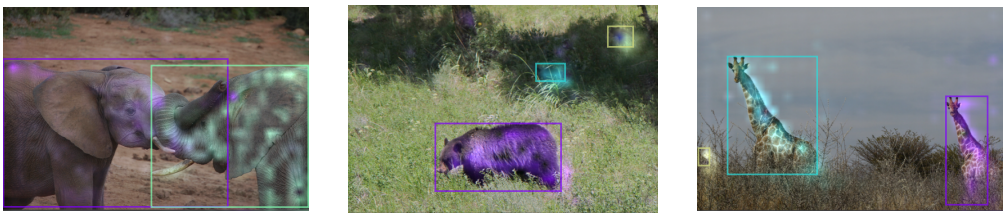


Figure 14: Examples of the model's predictions (at the score threshold of 0.5). Original images accessed by clicking the images in supported PDF readers.



(a) (b) (c)

图13：Transformer解码器在给定边界框条件下（预测类别标记时）的交叉注意力可视化。

检测结果可视化

在图14中，我们可视化展示了Pix2seq模型（AP值为46）在COCO验证集部分图像上的检测结果，这些图像包含密集排列的物体。

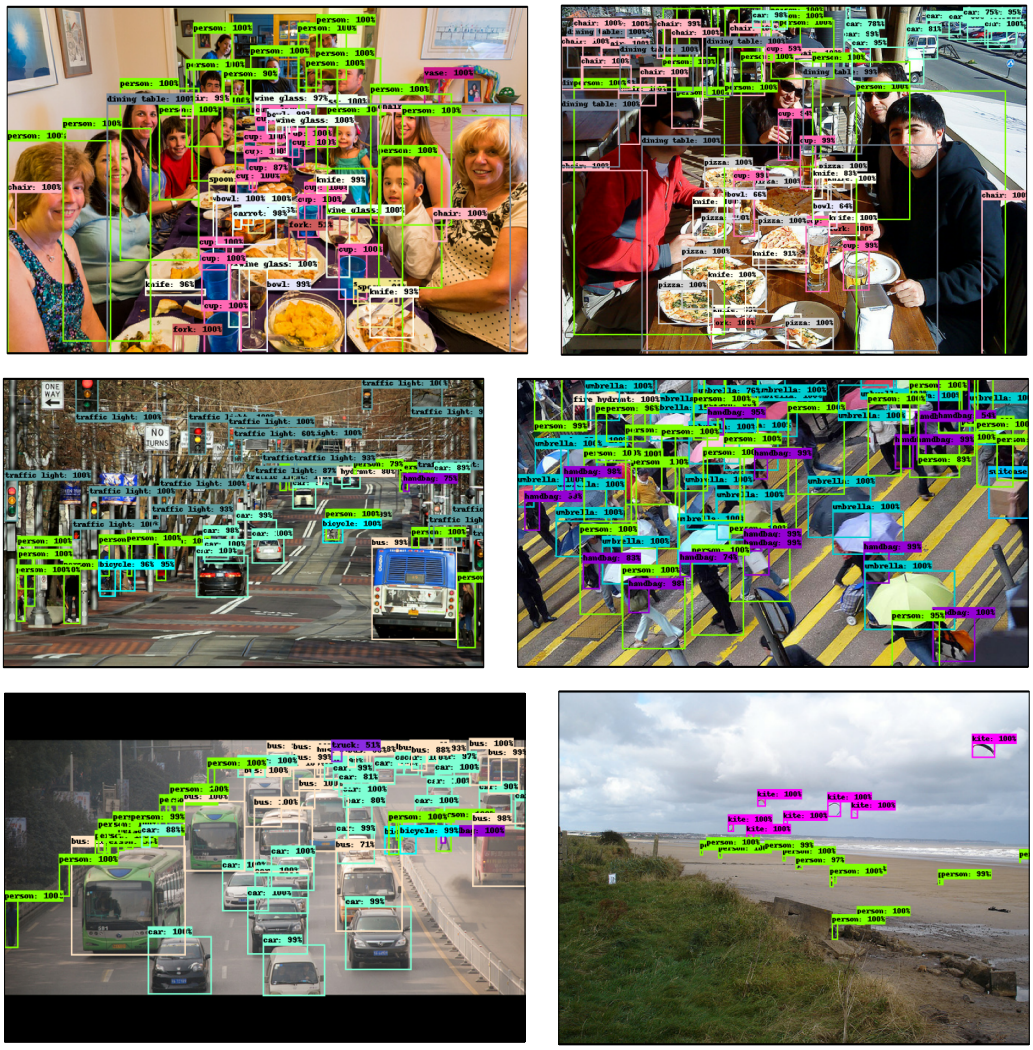


图14：模型预测示例（分数阈值为0.5）。通过点击支持的PDF阅读器中的图像可访问原始图像。