You Only Look at One Sequence: Rethinking Transformer in Vision through Object Detection

Yuxin Fang [1*] **Bencheng Liao** [1*] **Xinggang Wang** [1†] **Jiemin Fang** [2,1]
Jiyang Qi [1] **Rui Wu** [3] **Jianwei Niu** [3] **Wenyu Liu** [1]

[1] School of EIC, Huazhong University of Science & Technology
[2] Institute of AI, Huazhong University of Science & Technology
[3] Horizon Robotics
{yxf, bcliao, xgwang}@hust.edu.cn

Abstract

Can Transformer perform 2D object- and region-level recognition from a pure sequence-to-sequence perspective with minimal knowledge about the 2D spatial structure? To answer this question, we present You Only Look at One Sequence (YOLOS), a series of object detection models based on the vanilla Vision Transformer with the fewest possible modifications, region priors, as well as inductive biases of the target task. We find that YOLOS pre-trained on the mid-sized ImageNet-1k dataset *only* can already achieve quite competitive performance on the challenging COCO object detection benchmark, *e.g.*, YOLOS-Base directly adopted from BERT-Base architecture can obtain 42.0 box AP on COCO `val`. We also discuss the impacts as well as limitations of current pre-train schemes and model scaling strategies for Transformer in vision through YOLOS. Code and pre-trained models are available at https://github.com/hustvl/YOLOS.

1 Introduction

Transformer [58] is born to transfer. In natural language processing (NLP), the dominant approach is to first pre-train Transformer on large, generic corpora for general language representation learning, and then fine-tune or adapt the model on specific target tasks [18]. Recently, Vision Transformer (ViT) [1] [21] demonstrates that canonical Transformer encoder architecture directly inherited from NLP can perform surprisingly well on image recognition at scale using modern vision transfer learning recipe [33]. Taking sequences of image patch embeddings as inputs, ViT can successfully transfer pre-trained general visual representations from sufficient scale to more specific image classification tasks with fewer data points from a pure sequence-to-sequence perspective.

Since a pre-trained Transformer can be successfully fine-tuned on sentence-level tasks [7, 19] in NLP, as well as *token-level* tasks [48, 52], where models are required to produce fine-grained output at the token-level [18]. A natural question is: Can ViT transfer to more challenging *object- and region-level* target tasks in computer vision such as object detection other than image-level recognition?

ViT-FRCNN [6] is the first to use a pre-trained ViT as the backbone for a Faster R-CNN [50] object detector. However, this design cannot get rid of the reliance on convolutional neural networks (CNNs)

*Yuxin Fang and Bencheng Liao contributed equally. †Xinggang Wang is the corresponding author. This work was done when Yuxin Fang was interning at Horizon Robotics mentored by Rui Wu.

[1]There are various sophisticated or hybrid architectures termed as "Vision Transformer". For disambiguation, in this paper, "Vision Transformer" and "ViT" refer to the canonical or vanilla Vision Transformer architecture proposed by Dosovitskiy et al. [21] unless specified.

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

仅观一序列：通过目标检测重新思考视觉中的Transformer

方玉新 [1*] 廖本成 [1*] 王兴刚 [1†] 方杰敏 [2,1] 齐继阳 [1] 吴锐 [3] 牛建伟 [3] 刘文予 [1]

1 华中科技大学电子信息与通信学院 [2] 华中科技大学人工智能学院 [3] 地平线机器人 {yxf, bcliao, xgwang}@hust.edu.cn

摘要

Transformer能否仅从纯序列到序列的角度，以对二维空间结构的最小化认知，实现二维物体及区域级别的识别？为解答这一问题，我们提出了“你只需看一个序列”（YOLOS）——一系列基于原始视觉Transformer且改动最少、区域先验最少以及目标任务归纳偏置最少的目标检测模型。我们发现，仅在中型ImageNet-1*k*数据集*only*上预训练的YOLOS，在极具挑战性的COCO目标检测基准测试中已能取得相当有竞争力的表现*e.g.*。例如，直接采用BERT-Base架构的YOLOS-Base可在COCO验证集上获得42.0的框AP。我们还通过YOLOS探讨了当前预训练方案及Transformer在视觉领域的模型缩放策略的影响与局限。代码与预训练模型详见https://github.com/hustvl/YOLOS。

1 引言

Transformer [58] 生而为迁移。在自然语言处理（NLP）领域，主流方法是先在大型通用语料库上预训练Transformer以学习通用语言表征，再针对特定目标任务进行微调或适配[18]。近期，视觉Transformer（ViT）[1] [21] 证明，直接继承自NLP的标准Transformer编码器架构，采用现代视觉迁移学习方案[33]，能在大规模图像识别任务中表现出惊人性能。ViT以图像块嵌入序列作为输入，从纯粹序列到序列的视角，成功将预训练的通用视觉表征从足够规模迁移至数据点较少的特定图像分类任务。

由于预训练的Transformer模型已能成功微调于NLP中的句子级任务[7,19]，以及需要模型在标记级别生成细粒度输出的*token-level*任务[48,52][18]，一个自然产生的问题是：ViT能否迁移至计算机视觉中更具挑战性的*object- and region-level*目标任务，例如超越图像级识别的物体检测？

ViT-FRCNN [6] 是首个采用预训练ViT作为Faster R-CNN [50]目标检测器主干网络的方法。然而，该设计仍无法摆脱对卷积神经网络（CNNs）的依赖。

*Yuxin Fang and Bencheng Liao contributed equally. †Xinggang Wang is the corresponding author. This work was done when Yuxin Fang was interning at Horizon Robotics mentored by Rui Wu.

[1]There are various sophisticated or hybrid architectures termed as “Vision Transformer”. For disambiguation, in this paper, “Vision Transformer” and “ViT” refer to the canonical or vanilla Vision Transformer architecture proposed by Dosovitskiy et al. [21] unless specified.

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

and strong 2D inductive biases, as ViT-FRCNN re-interprets the output sequences of ViT to 2D spatial feature maps and depends on region-wise pooling operations (*i.e.,* RoIPool [23, 25] or RoIAlign [27]) as well as region-based CNN architectures [50] to decode ViT features for object- and region-level perception. Inspired by modern CNN design, some recent works [39, 59, 62, 65] introduce the pyramidal feature hierarchy, spatial locality, equivariant as well as invariant representations [24] to canonical Vision Transformer design, which largely boost the performance in dense prediction tasks including object detection. However, these architectures are performance-oriented and cannot reflect the properties of the canonical or vanilla Vision Transformer [21] directly inherited from Vaswani et al. [58]. Another series of work, the DEtection TRansformer (DETR) families [10, 72], use a random initialized Transformer to encode & decode CNN features for object detection, which does not reveal the transferability of a pre-trained Transformer.

Intuitively, ViT is designed to model long-range dependencies and global contextual information instead of local and region-level relations. Moreover, ViT lacks hierarchical architecture as modern CNNs [26, 35, 53] to handle the large variations in the scale of visual entities [1, 37]. Based on the available evidence, it is still unclear whether a pure ViT can transfer pre-trained general visual representations from image-level recognition to the much more complicated 2D object detection task.

To answer this question, we present You Only Look at One Sequence (YOLOS), a series of object detection models based on the canonical ViT architecture with the fewest possible modifications, region priors, as well as inductive biases of the target task injected. Essentially, the change from a pre-trained ViT to a YOLOS detector is embarrassingly simple: (1) YOLOS replaces one [CLS] token for image classification in ViT with one hundred [DET] tokens for object detection. (2) YOLOS replaces the image classification loss in ViT with the bipartite matching loss to perform object detection in a set prediction manner following Carion et al. [10], which can avoid re-interpreting the output sequences of ViT to 2D feature maps as well as prevent manually injecting heuristics and prior knowledge of object 2D spatial structure during label assignment [71]. Moreover, the prediction head of YOLOS can get rid of complex and diverse designs, which is as compact as a classification layer.

Directly inherited from ViT [21], YOLOS is not designed to be yet another high-performance object detector, but to unveil the versatility and transferability of pre-trained canonical Transformer from image recognition to the more challenging object detection task. Concretely, our main contributions are summarized as follows:

- We use the mid-sized ImageNet-1k [51] as the *sole* pre-training dataset, and show that a vanilla ViT [21] can be successfully transferred to perform the complex object detection task and produce competitive results on COCO [36] benchmark with the fewest possible modifications, *i.e.*, by only looking at one sequence (YOLOS).
- For the first time, we demonstrate that 2D object detection can be accomplished in a pure sequence-to-sequence manner by taking a sequence of fixed-sized non-overlapping image patches as input. Among existing object detectors, YOLOS utilizes the minimal 2D inductive biases.
- For the vanilla ViT, we find the object detection results are quite sensitive to the pre-train scheme and the detection performance is far from saturating. Therefore the proposed YOLOS can be also used as a challenging benchmark task to evaluate different (label-supervised and self-supervised) pre-training strategies for ViT.

2 You Only Look at One Sequence

As for the model design, YOLOS closely follows the original ViT architecture [21], and is optimized for object detection in the same vein as Carion et al. [10]. YOLOS can be easily adapted to various canonical Transformer architectures available in NLP as well as in computer vision. This intentionally simple setup is not designed for better detection performance, but to exactly reveal characteristics of the Transformer family in object detection as unbiased as possible.

2.1 Architecture

An overview of the model is depicted in Fig. 1. Essentially, the change from a ViT to a YOLOS detector is simple: (1) YOLOS drops the [CLS] token for image classification and appends one

以及强大的二维归纳偏置，因为ViT-FRCNN将ViT的输出序列重新解释为二维空间特征图，并依赖于区域池化操作（*i.e.,* RoIPool [23, 25]或RoIAlign [27]）以及基于区域的CNN架构[50]来解码ViT特征，以实现对象和区域级别的感知。受现代CNN设计的启发，近期一些研究[39, 59, 62, 65]在标准视觉Transformer设计中引入了金字塔特征层次、空间局部性、等变及不变表示[24]，这极大地提升了包括目标检测在内的密集预测任务的性能。然而，这些架构以性能为导向，无法直接反映源自Vaswani等人[58]的标准或原始视觉Transformer[21]的特性。另一系列工作，即DEtection TRansformer（DETR）家族[10, 72]，使用随机初始化的Transformer对CNN特征进行编码和解码以进行目标检测，这并未揭示预训练Transformer的可迁移性。

直观上，ViT旨在建模长距离依赖和全局上下文信息，而非局部及区域层面的关系。此外，ViT缺乏现代CNN[26,35,53]那样的层次化架构，难以应对视觉实体尺度上的巨大变化[1,37]。现有证据表明，尚不明确纯ViT能否将预训练的通用视觉表示从图像级识别迁移至更为复杂的2D目标检测任务。

为回答这一问题，我们提出了"仅观察单一序列"(YOLOS)——这一系列目标检测模型基于标准ViT架构，仅需最少程度的修改，无需区域先验知识，也不注入目标任务的归纳偏置。本质上，从预训练ViT到YOLOS检测器的转变简单得出奇：(1) YOLOS将ViT中用于图像分类的一个`CLS`标记替换为一百个`DET`标记以进行目标检测；(2) YOLOS采用Carion等人[10]提出的集合预测方式，用二分匹配损失替代ViT中的图像分类损失，这样既避免了将ViT输出序列重新解释为二维特征图，也防止了在标签分配时手动注入关于物体二维空间结构的启发式规则和先验知识[71]。此外，YOLOS的预测头可以摆脱复杂多样的设计，其简洁程度堪比分类层。

YOLOS直接继承自ViT[21]，其设计初衷并非成为又一个高性能目标检测器，而是为了揭示预训练标准Transformer从图像识别领域迁移至更具挑战性的目标检测任务时的通用性与可迁移性。具体而言，我们的主要贡献可归纳如下：

- 我们采用中等规模的ImageNet-1*k* [51]作为*sole*预训练数据集，结果表明，仅需最少量的修改（*i.e.*），通过观察单一序列（YOLOS），标准ViT模型[21]即可成功迁移至复杂的目标检测任务，并在COCO[36]基准测试中取得具有竞争力的结果。
- 我们首次证明，通过将一系列固定大小的非重叠图像块作为输入，可以以纯序列到序列的方式完成2D目标检测。在现有目标检测器中，YOLOS利用了最少的2D归纳偏差。
- 对于原始ViT，我们发现目标检测结果对预训练方案相当敏感，且检测性能远未达到饱和。因此，所提出的YOLOS也可作为一个具有挑战性的基准任务，用于评估ViT的不同（标签监督与自监督）预训练策略。

2 你只需看一个序列

至于模型设计，YOLOS严格遵循原始ViT架构[21]，并沿袭Carion等人[10]的思路针对目标检测任务进行了优化。YOLOS能轻松适配自然语言处理及计算机视觉领域各类标准Transformer架构。这种刻意简化的配置并非旨在提升检测性能，而是为了尽可能无偏差地揭示Transformer家族在目标检测中的本质特性。

2.1 架构

模型概览如图1所示。本质上，从ViT转变为YOLOS检测器的过程十分简单：(1) YOLOS舍弃了用于图像分类的[`CLS`]标记，并新增了一个

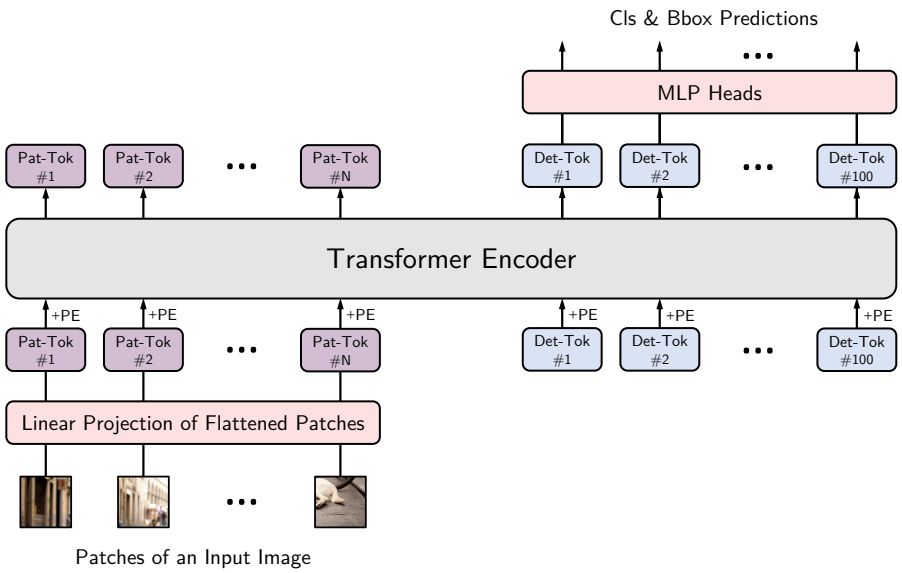

Figure 1: YOLOS architecture overview. "Pat-Tok" refers to [PATCH] token, which is the embedding of a flattened image patch. "Det-Tok" refers to [DET] token, which is a learnable embedding for object binding. "PE" refers to positional embedding. During training, YOLOS produces an optimal bipartite matching between predictions from one hundred [DET] tokens and ground truth objects. During inference, YOLOS directly outputs the final set of predictions in parallel. The figure style is inspired by Dosovitskiy et al. [21].

hundred randomly initialized learnable detection tokens ([DET] tokens) to the input patch embeddings ([PATCH] tokens) for object detection. (2) During training, YOLOS replaces the image classification loss in ViT with the bipartite matching loss to perform object detection in a set prediction manner following Carion et al. [10].

Stem. The canonical ViT [21] receives an 1D sequence of embedded tokens as the input. To handle 2D image inputs, we reshape the image $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ into a sequence of flattened 2D image patches $\mathbf{x}_{\texttt{PATCH}} \in \mathbb{R}^{N \times \left(P^2 \cdot C\right)}$. Here, (H, W) is the resolution of the input image, C is the number of input channels, (P, P) is the resolution of each image patch, and $N = \frac{HW}{P^2}$ is the resulting number of patches. Then we map $\mathbf{x}_{\texttt{PATCH}}$ to D dimensions with a trainable linear projection $\mathbf{E} \in \mathbb{R}^{\left(P^2 \cdot C\right) \times D}$. We refer to the output of this projection $\mathbf{x}_{\texttt{PATCH}}\mathbf{E}$ as [PATCH] tokens. Meanwhile, one hundred randomly initialized learnable [DET] tokens $\mathbf{x}_{\texttt{DET}} \in \mathbb{R}^{100 \times D}$ are appended to the [PATCH] tokens. Position embeddings $\mathbf{P} \in \mathbb{R}^{(N+100) \times D}$ are added to all the input tokens to retain positional information. We use the standard learnable 1D position embeddings following Dosovitskiy et al. [21]. The resulting sequence $\mathbf{z}_0$ serves as the input of YOLOS Transformer encoder. Formally:

$$\mathbf{z}_0 = \left[\mathbf{x}^1_{\texttt{PATCH}}\mathbf{E}; \cdots; \mathbf{x}^N_{\texttt{PATCH}}\mathbf{E};\ \mathbf{x}^1_{\texttt{DET}}; \cdots; \mathbf{x}^{100}_{\texttt{DET}}\right] + \mathbf{P}. \tag{1}$$

Body. The body of YOLOS is basically the same as ViT, which consists of a stack of Transformer encoder layers only [58]. [PATCH] tokens and [DET] tokens are treated equally and they perform global interactions inside Transformer encoder layers.

Each Transformer encoder layer consists of one multi-head self-attention (MSA) block and one MLP block. LayerNorm (LN) [2] is applied before every block, and residual connections [26] are applied after every block [3, 61]. The MLP contains one hidden layer with an intermediate GELU [29] non-linearity activation function. Formally, for the ℓ-*th* YOLOS Transformer encoder layer:

$$\begin{aligned} \mathbf{z}'_\ell &= \mathrm{MSA}\left(\mathrm{LN}\left(\mathbf{z}_{\ell-1}\right)\right) + \mathbf{z}_{\ell-1}, \\ \mathbf{z}_\ell &= \mathrm{MLP}\left(\mathrm{LN}\left(\mathbf{z}'_\ell\right)\right) + \mathbf{z}'_\ell. \end{aligned} \tag{2}$$

Detector Heads. The detector head of YOLOS gets rid of complex and heavy designs, and is as neat as the image classification layer of ViT. Both the classification and the bounding box regression heads are implemented by one MLP with separate parameters containing two hidden layers with intermediate ReLU [41] non-linearity activation functions.

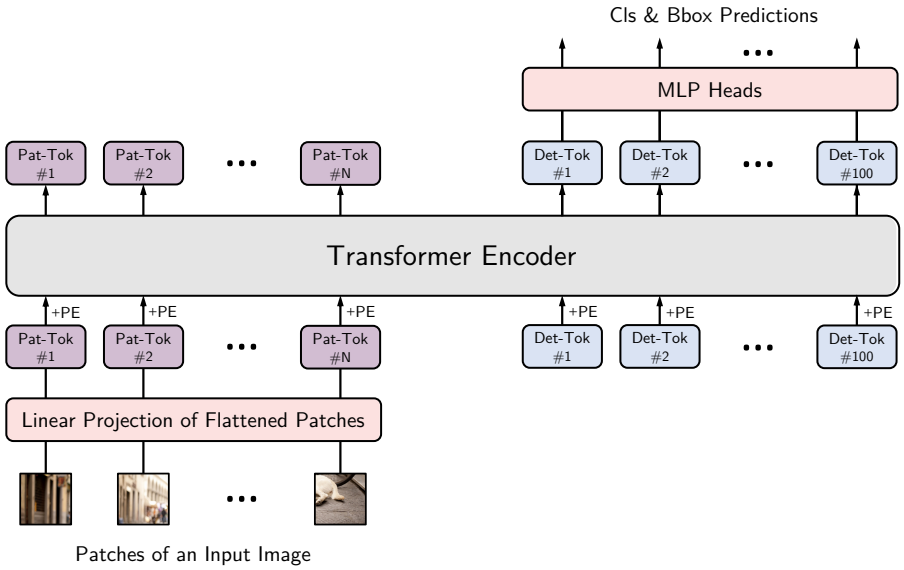

图1：YOLOS架构概览。"Pat-Tok"指[PATCH]令牌，代表扁平化图像块的嵌入。"Det-Tok"指[DET]令牌，是一个可学习的对象绑定嵌入。"PE"表示位置嵌入。训练期间，YOLOS在来自一百个[DET]令牌的预测与真实对象之间产生最优二分匹配。推理时，YOLOS直接并行输出最终的预测集。该图风格受Dosovitskiy等人[21]启发。

将一百个随机初始化的可学习检测标记（[DET]标记）添加到输入补丁嵌入（[PATCH]标记）中以进行目标检测。（2）在训练过程中，YOLOS遵循Carion等人[10]的方法，用二分匹配损失替换ViT中的图像分类损失，以集合预测的方式执行目标检测。

主干网络。标准的ViT[21]接收一维嵌入令牌序列作为输入。为处理二维图像输入，我们将图像$\mathbf{x} \in \mathbb{R}^{H \times W \times C}$重塑为展平的二维图像块序列$\mathbf{x}_{\texttt{PATCH}} \in \mathbb{R}^{N \times \left(P^2 \cdot C\right)}$。其中$(H, W)$表示输入图像的分辨率，$C$为输入通道数，$(P, P)$是每个图像块的分辨率，$N = \frac{HW}{P^2}$则是最终得到的块数量。随后通过可训练的线性投影$\mathbf{E} \in \mathbb{R}^{\left(P^2 \cdot C\right) \times D}$将$\mathbf{x}_{\texttt{PATCH}}$映射到$D$维空间。我们将该投影的输出$\mathbf{x}_{\texttt{PATCH}}\mathbf{E}$称为[PATCH]令牌。同时，向[PATCH]令牌后追加100个随机初始化的可学习[DET]令牌$\mathbf{x}_{\texttt{DET}} \in \mathbb{R}^{100 \times D}$。为保留位置信息，所有输入令牌都会加入位置嵌入$\mathbf{P} \in \mathbb{R}^{(N+100) \times D}$。我们采用Dosovitskiy等人[21]提出的标准可学习一维位置嵌入方法。最终形成的序列$\mathbf{z}_0$将作为YOLOS Transformer编码器的输入。形式化表示为：

$$\mathbf{z}_0 = \left[\mathbf{x}^1_{\texttt{PATCH}}\mathbf{E}; \cdots; \mathbf{x}^N_{\texttt{PATCH}}\mathbf{E};\ \mathbf{x}^1_{\texttt{DET}}; \cdots; \mathbf{x}^{100}_{\texttt{DET}}\right] + \mathbf{P}. \tag{1}$$

主体。YOLOS的主体与ViT基本相同，仅由一系列Transformer编码器层构成[58]。PATCH标记和DET标记被同等对待，它们在Transformer编码器层内部进行全局交互。

每个Transformer编码器层由一个多头自注意力（MSA）模块和一个MLP模块组成。在每个模块前应用LayerNorm（LN）[2]，每个模块后应用残差连接[26][3,61]。MLP包含一个具有中间GELU[29]非线性激活函数的隐藏层。形式上，对于{v*} YOLOS Transformer编码器层：

$$\begin{aligned} \mathbf{z}'_\ell &= \mathrm{MSA}\left(\mathrm{LN}\left(\mathbf{z}_{\ell-1}\right)\right) + \mathbf{z}_{\ell-1}, \\ \mathbf{z}_\ell &= \mathrm{MLP}\left(\mathrm{LN}\left(\mathbf{z}'_\ell\right)\right) + \mathbf{z}'_\ell. \end{aligned} \tag{2}$$

检测头。YOLOS的检测头摒弃了复杂且笨重的设计，其简洁程度与ViT的图像分类层相当。分类头和边界框回归头均由一个多层感知机（MLP）实现，该MLP拥有独立参数，包含两个隐藏层，中间采用ReLU[41]非线性激活函数。

Detection Token. We purposefully choose randomly initialized [DET] tokens as proxies for object representations to avoid inductive biases of 2D structure and prior knowledge about the task injected during label assignment. When fine-tuning on COCO, for each forward pass, an optimal bipartite matching between predictions generated by [DET] tokens and ground truth objects is established. This procedure plays the same role as label assignment [10, 71], but is unaware of the input 2D structure, *i.e.*, YOLOS does not need to re-interpret the output sequence of ViT to an 2D feature maps for label assignment. Theoretically, it is feasible for YOLOS to perform any dimensional object detection without knowing the exact spatial structure and geometry, as long as the input is always flattened to a sequence in the same way for each pass.

Fine-tuning at Higher Resolution. When fine-tuning on COCO, all the parameters are initialized from ImageNet-1k pre-trained weights except for the MLP heads for classification & bounding box regression as well as one hundred [DET] tokens, which are randomly initialized. During fine-tuning, the image has a much higher resolution than pre-training. We keep the patch size P unchanged, *i.e.*, $P \times P = 16 \times 16$, which results in a larger effective sequence length. While ViT can handle arbitrary input sequence lengths, the positional embeddings need to adapt to the longer input sequences with various lengths. We perform 2D interpolation of the pre-trained position embeddings on the fly[2].

Inductive Bias. We carefully design the YOLOS architecture for the minimal additional inductive biases injection. The inductive biases inherent from ViT come from the patch extraction at the network stem part as well as the resolution adjustment for position embeddings [21]. Apart from that, YOLOS adds no non-degenerated (*e.g.*, 3×3 or other non 1×1) convolutions upon ViT [3]. From the representation learning perspective, we choose to use [DET] tokens to bind objects for final predictions to avoid additional 2D inductive biases as well as task-specific heuristics. The performance-oriented design inspired by modern CNN architectures such as pyramidal feature hierarchy, 2D local spatial attention as well as the region-wise pooling operation is not applied. All these efforts are meant to exactly unveil the versatility and transferability of pre-trained Transformers from image recognition to object detection in a pure sequence-to-sequence manner, with minimal knowledge about the input spatial structure and geometry.

Comparisons with DETR. The design of YOLOS is deeply inspired by DETR [10]: YOLOS uses [DET] tokens following DETR as proxies for object representations to avoid inductive biases about 2D structures and prior knowledge about the task injected during label assignment, and YOLOS is optimized similarly as DETR.

Meanwhile, there are some key differences between the two models: (1) DETR adopts a Transformer encoder-decoder architecture, while YOLOS chooses an encoder-only Transformer architecture. (2) DETR only employs pre-training on its CNN backbone but leaves the Transformer encoder & decoder being trained from random initialization, while YOLOS naturally inherits representations from any pre-trained canonical ViT. (3) DETR applies cross-attention between encoded image features and object queries with auxiliary decoding losses deeply supervised at each decoder layer, while YOLOS always looks at only one sequence for each encoder layer, without distinguishing [PATCH] tokens and [DET] tokens in terms of operations. Quantitative comparisons between the two are in Sec. 3.4.

3 Experiments

3.1 Setup

Pre-training. We pre-train all YOLOS / ViT models on ImageNet-1k [51] dataset using the data-efficient training strategy suggested by Touvron et al. [57]. The parameters are initialized with a truncated normal distribution and optimized using AdamW [40]. The learning rate and batch size are 1×10^{-3} and 1024, respectively. The learning rate decay is cosine and the weight decay is 0.05. Rand-Augment [14] and random erasing [69] implemented by `timm` library [64] are used for data augmentation. Stochastic depth [32], Mixup [68] and Cutmix [66] are used for regularization.

[2] The configurations of position embeddings are detailed in the Appendix.

[3] We argue that it is imprecise to say Transformer do not have convolutions. All linear projection layers in Transformer are equivalent to point-wise or 1×1 convolutions with sparse connectivity, parameter sharing, and equivalent representations properties, which can largely improve the computational efficiency compared with the "all-to-all" interactions in fully-connected design that has even weaker inductive biases [5, 24].

检测标记。我们特意选择随机初始化的[DET]标记作为物体表征的代理，以避免在标签分配过程中引入的二维结构归纳偏置及任务先验知识。在COCO数据集上进行微调时，每次前向传播都会建立由[DET]标记生成的预测与真实物体之间的最优二分匹配。这一过程与标签分配[10,71]起着相同作用，但无需感知输入的二维结构*i.e.*——YOLOS无需将ViT的输出序列重新解释为二维特征图以进行标签分配。理论上，只要每次输入始终以相同方式展平为序列，YOLOS能够在不知晓具体空间结构和几何形状的情况下，执行任意维度的物体检测。

更高分辨率的微调。在COCO上进行微调时，除分类与边界框回归的MLP头部以及随机初始化的一百个[DET]标记外，所有参数均从ImageNet-1k预训练权重初始化。微调过程中，图像分辨率远高于预训练阶段。我们保持补丁大小P不变，即*i.e.*、$P \times P = 16 \times 16$，这导致有效序列长度增大。尽管ViT能处理任意输入序列长度，但位置嵌入需适应不同长度的更长输入序列。我们实时对预训练位置嵌入进行二维插值[2]。

归纳偏置。我们精心设计了YOLOS架构，以最小化额外归纳偏置的引入。源自ViT的固有偏置仅来自网络初始部分的图像块提取以及位置嵌入的分辨率调整[21]。除此之外，YOLOS未在ViT[3]基础上添加任何非退化（*e.g.*、3×3或其他非1×1）卷积操作。从表征学习的角度，我们选择使用[DET]标记来绑定对象以进行最终预测，从而避免引入额外的二维归纳偏置及任务特定启发式方法。受现代CNN架构启发的性能导向设计——如金字塔特征层级、二维局部空间注意力机制以及区域池化操作——均未被采用。所有这些努力旨在纯粹以序列到序列的方式，精确揭示预训练Transformer从图像识别到目标检测的通用性与可迁移性，同时最小化对输入空间结构和几何先验知识的依赖。

与DETR的对比。YOLOS的设计深受DETR[10]启发：YOLOS沿袭DETR采用{v*}标记作为物体表征的代理，以避免在标签分配过程中引入关于二维结构的归纳偏置及任务先验知识，其优化方式也与DETR类似。

与此同时，两种模型存在一些关键差异：(1) DETR采用Transformer编码器-解码器架构，而YOLOS选择了仅含编码器的Transformer架构。(2) DETR仅对其CNN骨干网络进行预训练，而Transformer编码器和解码器则从随机初始化开始训练；YOLOS则天然继承了任何预训练标准ViT的表征能力。(3) DETR在编码后的图像特征与对象查询之间应用交叉注意力，并通过辅助解码损失对每个解码器层进行深度监督；而YOLOS每个编码器层始终只处理单一序列，在操作层面不区分{v*}令牌与{v*}令牌。二者量化对比详见第3.4节。

3 实验

3.1 设置

预训练。我们采用Touvron等人[57]提出的数据高效训练策略，在ImageNet-1k[51]数据集上对所有YOLOS/ViT模型进行预训练。参数初始化采用截断正态分布，并使用AdamW[40]进行优化。学习率和批量大小分别为$1\times\times10^{-3}$和1024。学习率衰减采用余弦调度，权重衰减为0.05。数据增强采用timm库[64]实现的Rand-Augment[14]和随机擦除[69]。正则化方法包括随机深度[32]、Mixup[68]和Cutmix[66]。

位置嵌入的具体配置详见附录。[3]我们认为，简单地认为Transformer不含卷积操作是不准确的。Transformer中的所有线性投影层实际上等同于点积或1×1卷积，具备稀疏连接、参数共享及等效表示特性，相比全连接设计中“全对全”交互（其归纳偏置更弱）[5,24]，这些特性可大幅提升计算效率。

Fine-tuning. We fine-tune all YOLOS models on COCO object detection benchmark [36] in a similar way as Carion et al. [10]. All the parameters are initialized from ImageNet-1k pre-trained weights except for the MLP heads for classification & bounding box regression as well as one hundred [DET] tokens, which are randomly initialized. We train YOLOS on a single node with 8×12G GPUs. The learning rate and batch sizes are 2.5×10^{-5} and 8 respectively. The learning rate decay is cosine and the weight decay is 1×10^{-4}.

As for data augmentation, we use multi-scale augmentation, resizing the input images such that the shortest side is at least 256 and at most 608 pixels while the longest at most 864 for tiny models. For small and base models, we resize the input images such that the shortest side is at least 480 and at most 800 pixels while the longest at most 1333. We also apply random crop augmentations during training following Carion et al. [10]. The number of [DET] tokens are 100 and we keep the loss function as well as loss weights the same as DETR, while we don't apply dropout [54] or stochastic depth during fine-tuning since we find these regularization methods hurt performance.

Model Variants. With available computational resources, we study several YOLOS variants. Detailed configurations are summarized in Tab. 1. The input patch size for all models is 16×16. YOLOS-Ti (Tiny), -S (Small), and -B (Base) directly correspond to DeiT-Ti, -S, and -B [57]. From the model scaling perspective [20, 56, 60], the small and base models of YOLOS / DeiT can be seen as performing width scaling (w) [30, 67] on the corresponding tiny model.

Model	DeiT [57] Model	Layers (Depth)	Embed. Dim. (Width)	Pre-train Resolution	Heads	Params.	FLOPs	$\frac{f(\texttt{Lin.})}{f(\texttt{Att.})}$
YOLOS-Ti	DeiT-Ti	12	192	224	3	5.7 M	1.2 G	5.9
YOLOS-S	DeiT-S		384		6	22.1 M	4.5 G	11.8
YOLOS-B	DeiT-B		768		12	86.4 M	17.6 G	23.5
YOLOS-S (dwr)	–	19	240	272	6	13.7 M	4.6 G	5.0
YOLOS-S ($d\mathbf{w}r$)	–	14	330	240	6	19.0 M	4.6 G	8.8

Table 1: Variants of YOLOS. "dwr" and "$d\mathbf{w}r$" refer to uniform compound model scaling and fast model scaling, respectively. The "dwr" and "$d\mathbf{w}r$" notations are inspired by Dollár et al. [20]. Note that all the numbers listed are for pre-training, which could change during fine-tuning, *e.g.*, the resolution and FLOPs.

Besides, we investigate two other model scaling strategies which proved to be effective in CNNs. The first one is uniform compound scaling (dwr) [20, 56]. In this case, the scaling is uniform w.r.t. FLOPs along all model dimensions (*i.e.*, width (w), depth (d) and resolution (r)). The second one is fast scaling ($d\mathbf{w}r$) [20] that encourages primarily scaling model width ($\mathbf{w}$), while scaling depth (d) and resolution (r) to a lesser extent w.r.t. FLOPs. During the ImageNet-1k pre-training phase, we apply dwr and $d\mathbf{w}r$ scaling to DeiT-Ti (~ 1.2G FLOPs) and scale the model to ~ 4.5G FLOPs to align with the computations of DeiT-S. Larger models are left for future work.

For canonical CNN architectures, the model complexity or FLOPs (f) are proportional to dw^2r^2 [20]. Formally, $f(\texttt{CNN}) \propto dw^2r^2$. Different from CNN, there are two kinds of operations that contribute to the FLOPs of ViT. The first one is the linear projection (Lin.) or point-wise convolution, which fuses the information across different channels point-wisely via learnable parameters. The complexity is $f(\texttt{Lin.}) \propto dw^2r^2$, which is the same as $f(\texttt{CNN})$. The second one is the spatial attention (Att.), which aggregates the spatial information depth-wisely via computed attention weights. The complexity is $f(\texttt{Att.}) \propto dwr^4$, which grows quadratically with the input sequence length or number of pixels.

Note that the available scaling strategies are designed for architectures with complexity $f \propto dw^2r^2$, so theoretically the dwr as well as $d\mathbf{w}r$ model scaling are not directly applicable to ViT. However, during pre-training phase the resolution is relatively low, therefore $f(\texttt{Lin.})$ dominates the FLOPs ($\frac{f(\texttt{Lin.})}{f(\texttt{Att.})} > 5$). Our experiments indicate that some model scaling properties of ViT are consistent with CNNs when $\frac{f(\texttt{Lin.})}{f(\texttt{Att.})}$ is large.

3.2 The Effects of Pre-training

We study the effects of different pre-training strategies (both label-supervised and self-supervised) when transferring ViT (DeiT-Ti and DeiT-S) from ImageNet-1k to the COCO object detection benchmark via YOLOS. For object detection, the input shorter size is 512 for tiny models and is 800 for small models during inference. The results are shown in Tab. 2 and Tab. 3.

微调。我们以类似于Carion等人[10]的方式，在COCO目标检测基准[36]上对所有YOLOS模型进行微调。除分类与边界框回归的MLP头部以及一百个[DET]令牌（这些是随机初始化的）外，所有参数均从ImageNet-1k预训练权重初始化。我们在单节点上使用8块$\times$ 12G GPU训练YOLOS模型。学习率和批量大小分别为2.5×10^{-5}和8。学习率衰减采用余弦策略，权重衰减为1×10^{-4}。

在数据增强方面，我们采用多尺度增强方法，对输入图像进行尺寸调整：对于微型模型，最短边至少为256像素，至多608像素，而最长边不超过864像素；对于小型和基础模型，最短边至少480像素，至多800像素，最长边不超过1333像素。训练过程中，我们还按照Carion等人[10]的方法应用随机裁剪增强。[{v*}]标记的数量为100个，并保持与DETR相同的损失函数及损失权重。在微调阶段，我们不使用dropout[54]或随机深度等正则化方法，因为发现这些方法会损害模型性能。

模型变体。基于可用的计算资源，我们研究了多个YOLOS变体。详细配置总结于表1中。所有模型的输入补丁尺寸均为16×16。YOLOS-Ti（微型）、-S（小型）和-B（基础）直接对应DeiT-Ti、-S和-B[57]。从模型缩放的角度来看[20,56,60]，YOLOS/DeiT的小型和基础模型可视为在对应微型模型上执行宽度缩放w[30,67]。

Model	DeiT [57] Model	Layers (Depth)	Embed. Dim. (Width)	Pre-train Resolution	Heads	Params.	FLOPs	$\frac{f(\texttt{Lin.})}{f(\texttt{Att.})}$
YOLOS-Ti	DeiT-Ti		192		3	5.7 M	1.2 G	5.9
YOLOS-S	DeiT-S	12	384	224	6	22.1 M	4.5 G	11.8
YOLOS-B	DeiT-B		768		12	86.4 M	17.6 G	23.5
YOLOS-S (dwr)	–	19	240	272	6	13.7 M	4.6 G	5.0
YOLOS-S ($d\mathbf{w}r$)	–	14	330	240	6	19.0 M	4.6 G	8.8

表1：YOLOS的变体。“dwr”和“$d\mathbf{w}r$”分别指代统一复合模型缩放与快速模型缩放。“dwr”与“$d\mathbf{w}r$”的命名灵感源自Dollár等人[20]。请注意，所列数据均为预训练阶段数值，在微调过程中可能发生变化，*e.g.*，包括分辨率与浮点运算次数。

此外，我们还研究了另外两种在CNN中验证有效的模型缩放策略。第一种是均匀复合缩放（dwr）[20, 56]。该策略对所有模型维度（*i.e.*、宽度（w）、深度（d）和分辨率（r）进行与FLOPs成比例的均匀缩放。第二种是快速缩放（$d\mathbf{w}r$）[20]，该策略主要鼓励扩展模型宽度（$\mathbf{w}$），同时较小幅度地调整深度（d）和分辨率（r）以匹配FLOPs。在ImageNet-1k预训练阶段，我们对DeiT-Ti（$\sim$ 1.2G FLOPs）应用dwr和$d\mathbf{w}r$缩放，将模型扩展至$\sim$ 4.5G FLOPs以对齐DeiT-S的计算量。更大规模的模型留待后续研究。

对于经典的CNN架构，模型复杂度或FLOPs（f）与dw^2r^2成正比[20]。形式上表示为$f(\texttt{CNN}) \propto dw^2r^2$。与CNN不同，ViT的FLOPs由两类操作构成：首先是线性投影（Lin.）或逐点卷积，它通过可学习参数跨通道逐点融合信息，其复杂度为$f(\texttt{Lin.}) \propto dw^2r^2$，与$f(\texttt{CNN})$相同；其次是空间注意力（Att.），通过计算注意力权重沿深度方向聚合空间信息，其复杂度为$f(\texttt{Att.}) \propto dwr^4$，随输入序列长度或像素数量呈二次方增长。

需要注意的是，现有的缩放策略是为复杂度为$f \propto dw^2r^2$的架构设计的，因此理论上dwr和$d\mathbf{w}r$的模型缩放方法并不直接适用于ViT。然而，在预训练阶段，由于分辨率相对较低，$f(\texttt{Lin.})$主导了FLOPs（$\frac{f(\texttt{Lin.})}{f(\texttt{Att.})} > 5$）。我们的实验表明，当$\frac{f(\texttt{Lin.})}{f(\texttt{Att.})}$较大时，ViT的某些模型缩放特性与CNNs保持一致。

3.2 预训练的影响

我们研究了不同预训练策略（包括标签监督和自监督）在将ViT（DeiT-Ti和DeiT-S）从ImageNet-1k通过YOLOS迁移至COCO目标检测基准时的效果。对于目标检测任务，在推理阶段，微型模型的输入短边尺寸设为512，小型模型则为800。实验结果展示于表2和表3中。

Model	Pre-train Method	Pre-train Epochs	Fine-tune Epochs	Pre-train pFLOPs	Fine-tune pFLOPs	Total pFLOPs	ImNet Top-1	AP
YOLOS-Ti	Rand. Init.	0	600	0	14.2×10^2	14.2×10^2	–	19.7
	Label Sup. [57]	200	300	3.1×10^2	7.1×10^2	10.2×10^2	71.2	26.9
	Label Sup. [57]	300		4.7×10^2		11.8×10^2	72.2	28.7
	Label Sup. (⚗) [57]	300		4.7×10^2		11.8×10^2	74.5	29.7
YOLOS-S	Rand. Init.	0	250	0	5.9×10^3	5.9×10^3	–	20.9
	Label Sup. [57]	100	150	0.6×10^3	3.5×10^3	4.1×10^3	74.5	32.0
	Label Sup. [57]	200		1.2×10^3		4.7×10^3	78.5	36.1
	Label Sup. [57]	300		1.8×10^3		5.3×10^3	79.9	36.1
	Label Sup. (⚗) [57]	300		1.8×10^3		5.3×10^3	81.2	37.2

Table 2: The effects of label-supervised pre-training. "pFLOPs" refers to petaFLOPs ($\times 10^{15}$). "ImNet" refers to ImageNet-1k. "⚗" refers to the distillation method from Touvron et al. [57].

Model	Self Sup. Pre-train Method	Pre-train Epochs	Fine-tune Epochs	Linear Acc.	AP
YOLOS-S	MoCo-v3 [13]	300	150	73.2	33.6
	DINO [11]	800	150	77.0	36.2

Table 3: Study of self-supervised pre-training on YOLOS-S.

Necessity of Pre-training. At least under prevalent transfer learning paradigms [10, 57], the pre-training is necessary in terms of computational efficiency. For both tiny and small models, we find that pre-training on ImageNet-1k saves the total theoretical forward pass computations (total pre-training FLOPs & total fine-tuning FLOPs) compared with training on COCO from random initialization (training from scratch [28]). Models trained from scratch with hundreds of epochs still lag far behind the pre-trained ViT even if given more total FLOPs budgets. This seems quite different from canonical modern CNN-based detectors, which can catch up with pre-trained counterparts quickly [28].

Label-supervised Pre-training. For supervised pre-training with ImageNet-1k ground truth labels, we find that different-sized models prefer different pre-training schedules: 200 epochs pre-training for YOLOS-Ti still cannot catch up with 300 epochs pre-training even with a 300 epochs fine-tuning schedule, while for the small model 200 epochs pre-training provides feature representations as good as 300 epochs pre-training for transferring to the COCO object detection benchmark.

With additional transformer-specific distillation ("⚗") introduced by Touvron et al. [57], the detection performance is further improved by ~ 1 AP for both tiny and small models, in part because exploiting a CNN teacher [47] during pre-training helps ViT adapt to COCO better. It is also promising to directly leverage [DET] tokens to help smaller YOLOS learn from larger YOLOS on COCO during fine-tuning in a similar way as Touvron et al. [57], we leave it for future work.

Self-supervised Pre-training. The success of Transformer in NLP greatly benefits from large-scale self-supervised pre-training [18, 44, 45]. In vision, pioneering works [12, 21] train self-supervised Transformers following the masked auto-encoding paradigm in NLP. Recent works [11, 13] based on siamese networks show intriguing properties as well as excellent transferability to downstream tasks. Here we perform a preliminary transfer learning experiment on YOLOS-S using MoCo-v3 [13] and DINO [11] self-supervised pre-trained ViT weights in Tab. 3.

The transfer learning performance of 800 epochs DINO self-supervised model on COCO object detection is on a par with 300 epochs DeiT label-supervised pre-training, suggesting great potentials of self-supervised pre-training for ViT on challenging object-level recognition tasks. Meanwhile, the transfer learning performance of MoCo-v3 is less satisfactory, in part for the MoCo-v3 weight is heavily under pre-trained. Note that the pre-training epochs of MoCo-v3 are the same as DeiT (300 epochs), which means that there is still a gap between the current state-of-the-art self-supervised pre-training approach and the prevalent label-supervised pre-training approach for YOLOS.

YOLOS as a Transfer Learning Benchmark for ViT. From the above analysis, we conclude that the ImageNet-1k pre-training results cannot precisely reflect the transfer learning performance on COCO object detection. Compared with widely used image recognition transfer learning benchmarks such as CIFAR-10/100 [34], Oxford-IIIT Pets [43] and Oxford Flowers-102 [42], the performance of

Model	Pre-train Method	Pre-train Epochs	Fine-tune Epochs	Pre-train pFLOPs	Fine-tune pFLOPs	Total pFLOPs	ImNet Top-1	AP
YOLOS-Ti	Rand. Init.	0	600	0	14.2×10^2	14.2×10^2	–	19.7
	Label Sup. [57]	200	300	3.1×10^2	7.1×10^2	10.2×10^2	71.2	26.9
	Label Sup. [57]	300		4.7×10^2		11.8×10^2	72.2	28.7
	Label Sup. (⚗) [57]	300		4.7×10^2		11.8×10^2	74.5	29.7
YOLOS-S	Rand. Init.	0	250	0	5.9×10^3	5.9×10^3	–	20.9
	Label Sup. [57]	100	150	0.6×10^3	3.5×10^3	4.1×10^3	74.5	32.0
	Label Sup. [57]	200		1.2×10^3		4.7×10^3	78.5	36.1
	Label Sup. [57]	300		1.8×10^3		5.3×10^3	79.9	36.1
	Label Sup. (⚗) [57]	300		1.8×10^3		5.3×10^3	81.2	37.2

表2：标签监督预训练的效果。“pFLOPs”指petaFLOPs（$\times 10^{15}$）。“ImNet”指ImageNet-1k。“⚗”指Touvron等人[57]提出的蒸馏方法。

Model	Self Sup. Pre-train Method	Pre-train Epochs	Fine-tune Epochs	Linear Acc.	AP
YOLOS-S	MoCo-v3 [13]	300	150	73.2	33.6
	DINO [11]	800	150	77.0	36.2

表3：YOLOS-S上的自监督预训练研究。

预训练的必要性。至少在当前主流的迁移学习范式[10,57]下，从计算效率角度考量，预训练是不可或缺的。我们发现，无论是微型还是小型模型，相较于从零开始训练（随机初始化后在COCO上训练[28]），使用ImageNet-1k进行预训练能节省总体理论前向计算量（总预训练FLOPs与总微调FLOPs之和）。即使赋予更多总FLOPs预算，经过数百轮从零开始训练的模型仍远落后于预训练的ViT。这与典型的现代基于CNN的检测器形成鲜明对比——后者能快速追平预训练模型的性能[28]。

标签监督的预训练。在使用ImageNet-1k真实标签进行监督预训练时，我们发现不同规模的模型偏好不同的预训练周期：YOLOS-Ti模型即使经过300个周期的微调，其200个周期的预训练表现仍无法赶上300个周期的预训练效果；而对于小型模型，200个周期的预训练所提供的特征表示在迁移至COCO目标检测基准时，与300个周期的预训练效果相当。

随着Touvron等人[57]引入的额外针对Transformer的蒸馏方法（“⚗”），检测性能通过$\sim$ 1 AP得到进一步提升，无论是微型还是小型模型均受益。这部分归功于预训练阶段采用CNN教师模型[47]帮助ViT更好地适应COCO数据集。同样具有前景的是直接利用[DET]标记，以类似Touvron等人[57]的方式，在微调阶段帮助较小型的YOLOS从较大型YOLOS学习COCO数据——我们将此方向留待未来研究。

自监督预训练。Transformer在NLP领域的成功很大程度上得益于大规模自监督预训练[18,44,45]。在视觉领域，开创性工作[12,21]遵循NLP中的掩码自编码范式训练自监督Transformer。近期基于孪生网络的研究[11,13]展现出引人入胜的特性以及出色的下游任务迁移能力。本文在YOLOS-S上使用MoCo-v3[13]和DINO[11]自监督预训练的ViT权重进行了初步迁移学习实验，结果如表3所示。

800轮DINO自监督模型在COCO目标检测上的迁移学习性能与300轮DeiT标签监督预训练相当，这表明自监督预训练在ViT应对挑战性物体级识别任务上具有巨大潜力。与此同时，MoCo-v3的迁移学习表现不尽如人意，部分原因在于MoCo-v3的权重严重欠预训练。需注意的是，MoCo-v3的预训练轮数与DeiT相同（均为300轮），这意味着当前最先进的自监督预训练方法与主流标签监督预训练方法在YOLOS上仍存在差距。

YOLOS作为ViT迁移学习的基准。通过上述分析，我们得出结论：ImageNet-1k预训练结果无法准确反映在COCO目标检测任务上的迁移学习性能。与广泛使用的图像识别迁移学习基准（如CIFAR-10/100 [34]、Oxford-IIIT Pets [43]和Oxford Flowers-102 [42]）相比，其表现

YOLOS on COCO is more sensitive to the pre-train scheme and the performance is far from saturating. Therefore it is reasonable to consider YOLOS as a challenging transfer learning benchmark to evaluate different (label-supervised or self-supervised) pre-training strategies for ViT.

3.3 Pre-training and Transfer Learning Performance of Different Scaled Models

We study the pre-training and the transfer learning performance of different model scaling strategies, *i.e.*, width scaling (w), uniform compound scaling (dwr) and fast scaling ($d\mathbf{w}r$). The models are scaled from $\sim$ 1.2G to $\sim$ 4.5G FLOPs regime for pre-training. Detailed model configurations and descriptions are given in Sec. 3.1 and Tab. 1.

We pre-train all the models for 300 epochs on ImageNet-1k with input resolution determined by the corresponding scaling strategies, and then fine-tune these models on COCO for 150 epochs. Few literatures are available for resolution scaling in object detection, where the inputs are usually oblong in shape and the multi-scale augmentation [10, 27] is used as a common practice. Therefore for each model during inference, we select the smallest resolution (*i.e.*, the shorter size) ranging in $[480, 800]$ producing the highest box AP, which is 784 for $d\mathbf{w}r$ scaling and 800 for all the others. The results are summarized in Tab. 4.

Scale	Image Classification @ ImageNet-1k				Object Detection @ COCO `val`			
	FLOPs	$\frac{f(\texttt{Lin.})}{f(\texttt{Att.})}$	FPS	Top-1	FLOPs	$\frac{f(\texttt{Lin.})}{f(\texttt{Att.})}$	FPS	AP
–	1.2 G	5.9	1315	72.2	81 G	0.28	12.0	29.6
w	4.5 G	11.8	615	79.9	194 G	0.55	5.7	36.1
dwr	4.6 G	5.0	386	80.5	163 G	0.35	4.5	36.2
$d\mathbf{w}r$	4.6 G	8.8	511	80.4	172 G	0.49	5.7	37.6

Table 4: Pre-training and transfer learning performance of different scaled models. FLOPs and FPS data of object detection are measured over the first 100 images of COCO `val` split during inference following Carion et al. [10]. FPS is measured with batch size 1 on a single 1080Ti GPU.

Pre-training. Both dwr and $d\mathbf{w}r$ scaling can improve the accuracy compared with simple w scaling, *i.e.*, the DeiT-S baseline. Other properties of each scaling strategy are also consistent with CNNs [20, 56], *e.g.*, w scaling is the most speed friendly. dwr scaling achieves the strongest accuracy. $d\mathbf{w}r$ is nearly as fast as w scaling and is on a par with dwr scaling in accuracy. Perhaps the reason why these CNN model scaling strategies are still appliable to ViT is that during pre-training the linear projection (1×1 convolution) dominates the model computations.

Transfer Learning. The picture changes when transferred to COCO. The input resolution r is much higher so the spatial attention takes over and linear projection part is no longer dominant in terms of FLOPs ($\frac{f(\texttt{Lin.})}{f(\texttt{Att.})} \propto \frac{w}{r^2}$). Canonical CNN model scaling recipes do not take spatial attention computations into account. Therefore there is some inconsistency between pre-training and transfer learning performance: Despite being strong on ImageNet-1k, the dwr scaling achieves similar box AP as simple w scaling. Meanwhile, the performance gain from $d\mathbf{w}r$ scaling on COCO cannot be clearly explained by the corresponding CNN scaling methodology that does not take $f(\texttt{Att.}) \propto dwr^4$ into account. The performance inconsistency between pre-training and transfer learning calls for novel model scaling strategies for ViT considering spatial attention complexity.

3.4 Comparisons with CNN-based Object Detectors

In previous sections, we treat YOLOS as a touchstone for the transferability of ViT. In this section, we consider YOLOS as an object detector and we compare YOLOS with some modern CNN detectors.

Comparisons with Tiny-sized CNN Detectors. As shown in Tab. 5, the tiny-sized YOLOS model achieves impressive performance compared with well-established and highly-optimized CNN object detectors. YOLOS-Ti is strong in AP and competitive in FLOPs & FPS even though Transformer is not intentionally designed to optimize these factors. From the model scaling perspective [20, 56, 60], YOLOS-Ti can serve as a promising model scaling start point.

Comparisons with DETR. The relations and differences in model design between YOLOS and DETR are given in Sec. 2.1, here we make quantitative comparisons between the two.

YOLOS在COCO数据集上对预训练方案更为敏感，其性能远未达到饱和。因此，将YOLOS视为一个具有挑战性的迁移学习基准，用于评估ViT的不同（标签监督或自监督）预训练策略是合理的。

3.3 不同规模模型的预训练与迁移学习性能

我们研究了不同模型缩放策略的预训练与迁移学习性能，包括*i.e.*、宽度缩放(w)、均匀复合缩放(dwr)及快速缩放($d\mathbf{w}r$)。模型在预训练阶段的计算量范围从∼1.2G扩展至∼4.5GFLOPs。具体模型配置与描述详见第3.1节及表1。

我们在ImageNet-1k上对所有模型进行了300个epoch的预训练，输入分辨率由相应的缩放策略决定，随后在COCO数据集上对这些模型进行了150个epoch的微调。关于目标检测中分辨率缩放的研究文献较少，该领域输入通常呈长方形，且多尺度增强[10,27]是普遍采用的方法。因此，在推理阶段，我们为每个模型在[480, 800]范围内选择能产生最高边界框AP的最小分辨率（*i.e.*，即较短边尺寸），其中$d\mathbf{w}r$缩放策略对应784，其他所有策略均采用800。具体结果汇总于表4。

Scale	Image Classification @ ImageNet-1k				Object Detection @ COCO `val`			
	FLOPs	$\frac{f(\texttt{Lin.})}{f(\texttt{Att.})}$	FPS	Top-1	FLOPs	$\frac{f(\texttt{Lin.})}{f(\texttt{Att.})}$	FPS	AP
–	1.2 G	5.9	1315	72.2	81 G	0.28	12.0	29.6
w	4.5 G	11.8	615	79.9	194 G	0.55	5.7	36.1
dwr	4.6 G	5.0	386	80.5	163 G	0.35	4.5	36.2
$d\mathbf{w}r$	4.6 G	8.8	511	80.4	172 G	0.49	5.7	37.6

表4：不同规模模型的预训练与迁移学习性能。目标检测的FLOPs和FPS数据是在推理过程中，按照Carion等人[10]的方法，在COCO `val`数据集的前100张图像上测得的。FPS是在单块1080Ti GPU上以批量大小1进行测量的。

预训练。与简单的w缩放（即DeiT-S基线*i.e.*）相比，dwr和$d\mathbf{w}r$两种缩放方式均能提升模型精度。各缩放策略的其他特性也与CNN模型[20,56]一致：*e.g.*、w缩放对计算速度最为友好，dwr缩放能实现最高精度，$d\mathbf{w}r$在速度上接近w缩放，精度表现则与dwr缩放相当。这些CNN模型缩放策略仍适用于ViT的原因可能在于：预训练阶段线性投影（1×1卷积）主导了模型计算量。

迁移学习。当转移到COCO数据集时，情况发生了变化。输入分辨率r更高，因此空间注意力占据主导地位，线性投影部分在FLOPs（$\frac{f(\texttt{Lin.})}{f(\texttt{Att.})} \propto \frac{w}{r^2}$）方面不再占优。传统的CNN模型缩放方法并未考虑空间注意力计算。这导致预训练与迁移学习性能之间存在不一致性：尽管在ImageNet-1k上表现强劲，dwr缩放实现的边界框AP与简单的w缩放相当。与此同时，$d\mathbf{w}r$缩放在COCO上的性能提升无法通过不考虑$f(\texttt{Att.}) \propto dwr^4$的相应CNN缩放方法论明确解释。预训练与迁移学习之间的性能差异，呼吁针对ViT考虑空间注意力复杂度的新型模型缩放策略。

3.4 与基于CNN的目标检测器比较

在前面的章节中，我们将YOLOS视为ViT可迁移性的试金石。本节中，我们将YOLOS作为目标检测器来考量，并将其与一些现代CNN检测器进行比较。

与微型CNN检测器的对比。如表5所示，与成熟且高度优化的CNN目标检测器相比，微型YOLOS模型展现出令人瞩目的性能。尽管Transformer并非专为优化这些因素而设计，YOLOS-Ti在平均精度（AP）上表现强劲，在浮点运算次数（FLOPs）和每秒帧率（FPS）方面也颇具竞争力。从模型缩放的角度来看[20,56,60]，YOLOS-Ti可作为一个有前景的模型缩放起点。

与DETR的对比。YOLOS与DETR在模型设计上的关联与差异已在第2.1节阐述，此处我们着重对两者进行量化比较。

Method	Backbone	Size	AP	Params. (M)	FLOPs (G)	FPS
YOLOv3-Tiny [49]	DarkNet [49]	416×416	16.6	8.9	5.6	330
YOLOv4-Tiny [60]	COSA [60]	416×416	21.7	6.1	7.0	371
YOLOS-Ti	DeiT-Ti (⚗) [57]	$256 \times *$	23.1	6.5	3.4	114
CenterNet [70]	ResNet-18 [26]	512×512	28.1	–	–	129
YOLOv4-Tiny (3l) [60]	COSA [60]	320×320	28.7	–	–	252
Def. DETR [72]	FBNet-V3 [15]	$800 \times *$	27.9	12.2	12.3	35
YOLOS-Ti	DeiT-Ti (⚗) [57]	$432 \times *$	28.6	6.5	11.7	84

Table 5: Comparisons with some tiny-sized modern CNN detectors. All models are trained to be fully converged. "Size" refers to input resolution for inference. FLOPs and FPS data are measured over the first 100 images of COCO `val` split during inference following Carion et al. [10]. FPS is measured with batch size 1 on a single 1080Ti GPU.

Method	Backbone	Epochs	Size	AP	Params. (M)	FLOPs (G)	FPS
Def. DETR [72]	FBNet-V3 [15]	150	$800 \times *$	27.5	12.2	12.3	35
YOLOS-Ti	DeiT-Ti [57]	300	$512 \times *$	28.7	6.5	18.8	60
YOLOS-Ti	DeiT-Ti (⚗) [57]	300	$432 \times *$	28.6	6.5	11.7	84
YOLOS-Ti	DeiT-Ti (⚗) [57]	300	$528 \times *$	30.0	6.5	20.7	51
DETR [10]	ResNet-18-DC5 [26]	150	$800 \times *$	36.9	29	129	7.4
YOLOS-S	DeiT-S [57]		$800 \times *$	36.1	31	194	5.7
YOLOS-S	DeiT-S (⚗) [57]		$800 \times *$	37.2	31	194	5.7
YOLOS-S ($d\mathbf{w}r$)	DeiT-S [57] ($d\mathbf{w}r$ Scale [20])		$704 \times *$	37.2	28	123	7.7
YOLOS-S ($d\mathbf{w}r$)	DeiT-S [57] ($d\mathbf{w}r$ Scale [20])		$784 \times *$	37.6	28	172	5.7
DETR [10]	ResNet-101-DC5 [26]	150	$800 \times *$	42.5	60	253	5.3
YOLOS-B	DeiT-B (⚗) [57]			42.0	127	538	2.7

Table 6: Comparisons with different DETR models. Tiny-sized models are trained to be fully converged. "Size" refers to input resolution for inference. FLOPs and FPS data are measured over the first 100 images of COCO `val` split during inference following Carion et al. [10]. FPS is measured with batch size 1 on a single 1080Ti GPU. The "ResNet-18-DC5" implantation is from `timm` library [64].

As shown in Tab. 6, YOLOS-Ti still performs better than the DETR counterpart, while larger YOLOS models with width scaling become less competitive: YOLOS-S with more computations is 0.8 AP lower compared with a similar-sized DETR model. Even worse, YOLOS-B cannot beat DETR with over $2\times$ parameters and FLOPs. Even though YOLOS-S with $d\mathbf{w}r$ scaling is able to perform better than the DETR counterpart, the performance gain cannot be clearly explained as discussed in Sec. 3.3.

Interpreting the Results. Although the performance is seemingly discouraging, the numbers are meaningful, as YOLOS is not purposefully designed for better performance, but designed to precisely reveal the transferability of ViT in object detection. *E.g.*, YOLOS-B is directly adopted from the BERT-Base architecture [18] in NLP. This 12 layers, 768 channels Transformer along with its variants have shown impressive performance on a wide range of NLP tasks. We demonstrate that with minimal modifications, this kind of architecture can also be successfully transferred (*i.e.*, AP $= 42.0$) to the challenging COCO object detection benchmark in computer vision from a pure sequence-to-sequence perspective. The minimal modifications from YOLOS exactly reveal the versatility and generality of Transformer.

3.5 Inspecting Detection Tokens

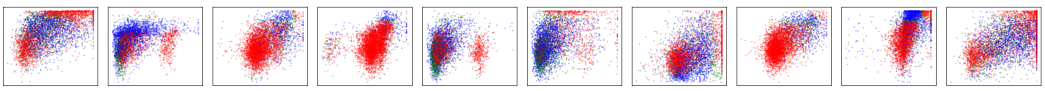

Figure 2: Visualization of all box predictions on all images from COCO `val` split for the first ten [`DET`] tokens. Each box prediction is represented as a point with the coordinates of its center normalized by each thumbnail image size. The points are color-coded so that **blue** points corresponds to small objects, **green** to medium objects and **red** to large objects. We observe that each [`DET`] token learns to specialize on certain regions and sizes. The visualization style is inspired by Carion et al. [10].

Method	Backbone	Size	AP	Params. (M)	FLOPs (G)	FPS
YOLOv3-Tiny [49]	DarkNet [49]	416×416	16.6	8.9	5.6	330
YOLOv4-Tiny [60]	COSA [60]	416×416	21.7	6.1	7.0	371
YOLOS-Ti	DeiT-Ti (⚗) [57]	$256 \times *$	23.1	6.5	3.4	114
CenterNet [70]	ResNet-18 [26]	512×512	28.1	–	–	129
YOLOv4-Tiny (3l) [60]	COSA [60]	320×320	28.7	–	–	252
Def. DETR [72]	FBNet-V3 [15]	$800 \times *$	27.9	12.2	12.3	35
YOLOS-Ti	DeiT-Ti (⚗) [57]	$432 \times *$	28.6	6.5	11.7	84

表5：与一些微型现代CNN检测器的比较。所有模型均训练至完全收敛。“尺寸”指推理时的输入分辨率。FLOPs和FPS数据是在推理过程中按照Carion等人[10]的方法，在COCO `val`数据集的前100张图像上测得的。FPS是在单块1080Ti GPU上以批量大小1进行测量的。

Method	Backbone	Epochs	Size	AP	Params. (M)	FLOPs (G)	FPS
Def. DETR [72]	FBNet-V3 [15]	150	$800 \times *$	27.5	12.2	12.3	35
YOLOS-Ti	DeiT-Ti [57]	300	$512 \times *$	28.7	6.5	18.8	60
YOLOS-Ti	DeiT-Ti (⚗) [57]	300	$432 \times *$	28.6	6.5	11.7	84
YOLOS-Ti	DeiT-Ti (⚗) [57]	300	$528 \times *$	30.0	6.5	20.7	51
DETR [10]	ResNet-18-DC5 [26]	150	$800 \times *$	36.9	29	129	7.4
YOLOS-S	DeiT-S [57]		$800 \times *$	36.1	31	194	5.7
YOLOS-S	DeiT-S (⚗) [57]		$800 \times *$	37.2	31	194	5.7
YOLOS-S ($d$wr)	DeiT-S [57] ($d$wr Scale [20])		$704 \times *$	37.2	28	123	7.7
YOLOS-S ($d$wr)	DeiT-S [57] ($d$wr Scale [20])		$784 \times *$	37.6	28	172	5.7
DETR [10]	ResNet-101-DC5 [26]	150	$800 \times *$	42.5	60	253	5.3
YOLOS-B	DeiT-B (⚗) [57]			42.0	127	538	2.7

表6：与不同DETR模型的对比。微型模型经过充分训练以达到完全收敛状态。“尺寸”指推理时的输入分辨率。FLOPs和FPS数据按照Carion等人[10]的方法，在推理阶段基于COCO `val`数据集前100张图像测得。FPS测量使用批大小为1，在单块1080Ti GPU上进行。“ResNet-18-DC5”实现来自timm库[64]。

如表6所示，YOLOS-Ti仍优于同规模的DETR模型，而通过宽度缩放构建的更大YOLOS模型则竞争力下降：计算量更大的YOLOS-S比体型相近的DETR模型低0.8 AP。更糟的是，参数量和计算量超出2×倍的YOLOS-B仍无法超越DETR。尽管采用dwr缩放的YOLOS-S性能优于对应DETR模型，但如第3.3节所述，这种性能优势尚无法明确解释。

解读结果。尽管性能表现看似不尽如人意，但这些数字实则意义重大，因为YOLOS并非为追求更高性能而设计，其初衷在于精确揭示ViT在目标检测任务中的可迁移性。*E.g.*YOLOS-B直接采用了自然语言处理领域BERT-Base架构[18]。这种12层、768通道的Transformer及其变体已在众多NLP任务中展现出卓越性能。我们证明，通过极简修改（*i.e.*AP值达=42.0），此类架构也能从纯序列到序列的视角，成功迁移至计算机视觉领域极具挑战性的COCO目标检测基准。YOLOS所采用的最小化改动恰恰彰显了Transformer架构的多功能性与普适性。

3.5 检测标记检查

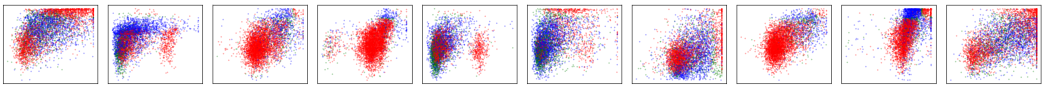

图2：在COCO `val`数据集的所有图像上，对前十个`[DET]`标记的所有边界框预测进行可视化。每个边界框预测以其中心坐标点表示，坐标已按各缩略图尺寸归一化。点按颜色编码：蓝色对应小物体，绿色对应中等物体，红色对应大物体。我们观察到每个`[DET]`标记会学习专注于特定区域和尺寸。该可视化风格受Carion等人[10]的启发。

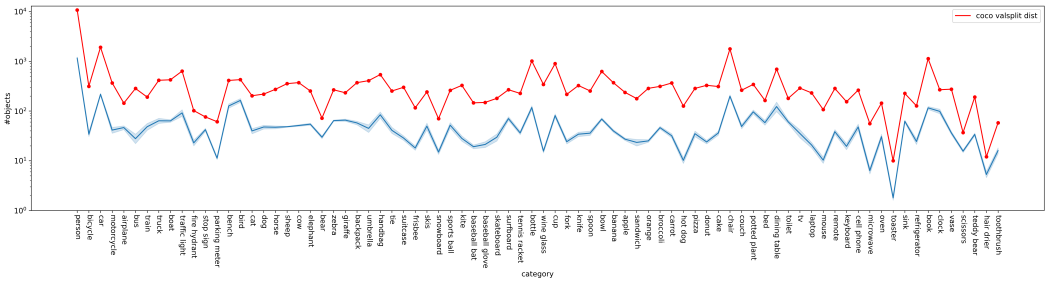

Figure 3: The statistics of all ground truth object categories (the **red** curve) and the statistics of all object category predictions from all [DET] tokens (the **blue** curve) on all images from COCO `val` split. The error bar of the **blue** curve represents the variability of the preference of different tokens for a given category, which is small. This suggests that different [DET] tokens are category insensitive.

Qualitative Analysis on Detection Tokens. As an object detector, YOLOS uses [DET] tokens to represent detected objects. In general, we find that [DET] tokens are sensitive to object locations and sizes, while insensitive to object categories, as shown in Fig. 2 and Fig. 3.

Quantitative Analysis on Detection Tokens. We give a quantitative analysis on the relation between X = the cosine similarity of [DET] token pairs, and Y = the corresponding predicted bounding box centers ℓ_2 distances. We use the Pearson correlation coefficient $\rho_{X,Y} = \frac{\mathbb{E}[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$ as a measure of linear correlation between variable X and Y, and we conduct this study on all predicted object pairs within each image in COCO `val` set averaged by all 5000 images. The result is $\rho_{X,Y} = -0.80$. This means that [DET] tokens that are close to each other (*i.e.*, with high cosine similarity) also lead to mostly nearby predictions (*i.e.*, with short ℓ_2 distances, given $\rho_{X,Y} < 0$).

We also conduct a quantitative study on the relation between X = the cosine similarity of [DET] token pairs, and Y = the corresponding cosine similarity of the output features of the classifier. The result is $\rho_{X,Y} = -0.07$, which is very close to 0. This means that there is no strong linear correlation between these two variables.

Detaching Detection Tokens. To further understand the role [DET] tokens plays, we study impacts caused by detaching the [DET] tokens of YOLOS during training, *i.e.*, we don't optimize the parameters of the one hundred randomly initialized [DET] tokens. As shown in Tab. 7, detaching the [DET] tokens has a minor impact to AP. These results imply that [DET] tokens mainly serve as the information carrier for the [PATCH] tokens. Similar phenomena are also observed in Fang et al. [22].

Model	[DET] Tokens Config	AP
YOLOS-Ti	Rand. Init. & Learnable	28.7
	Rand. Init. & **Detached**	28.3
YOLOS-S	Rand. Init. & Learnable	36.1
	Rand. Init. & **Detached**	36.4

Table 7: Impacts of detaching the [DET] tokens of YOLOS during training.

4 Related Work

Vision Transformer for Object Detection. There has been a lot of interest in combining CNNs with forms of self-attention mechanisms [4] to improve object detection performance [9, 31, 63], while recent works trend towards augmenting Transformer with CNNs (or CNN design). Beal et al. [6] propose to use a pre-trained ViT as the feature extractor for a Faster R-CNN [50] object detector. Despite being effective, they fail to ablate the CNN architectures, region-wise pooling operations [23, 25, 27] as well as hand-crafted components such as dense anchors [50] and NMS. Inspired by modern CNN architecture, some works [39, 59, 62, 65] introduce the pyramidal feature hierarchy and locality to Vision Transformer design, which largely boost the performance in dense prediction tasks including object detection. However, these architectures are performance-oriented and cannot reflect the properties of the canonical or vanilla Vision Transformer [21] that directly inherited from Vaswani et al. [58]. Another series of work, the DEtection TRansformer (DETR) families [10, 72], use a random initialized Transformer to encode & decode CNN features for object detection, which does not reveal the transferability of a pre-trained Transformer.

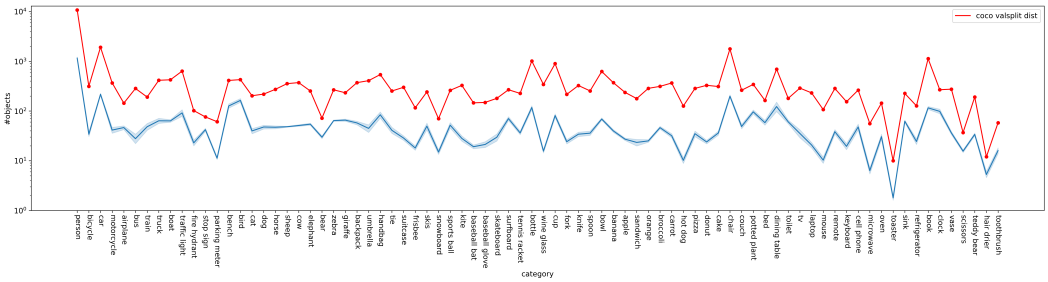

图3：COCO `val` 分割数据集中所有图像上真实物体类别的统计（红色曲线）与所有 [DET] 令牌预测的物体类别统计（蓝色曲线）对比。蓝色曲线的误差条表示不同令牌对特定类别偏好的变异性，这一数值较小，表明不同的 [DET] 令牌对类别不敏感。

检测令牌的定性分析。作为目标检测器，YOLOS使用[DET]令牌来表示检测到的对象。总体而言，我们发现[DET]令牌对目标位置和大小敏感，而对目标类别不敏感，如图2和图3所示。

检测令牌的定量分析。我们对X =令牌对[DET]的余弦相似度与Y =对应预测边界框中心ℓ_2距离之间的关系进行了定量分析。采用皮尔逊相关系数$\rho_{X,Y} = \frac{\mathbb{E}[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$衡量变量$X$与$Y$之间的线性相关性，该研究基于COCO验证集中每张图像内所有预测对象对的平均值（共5000张图像）。结果显示为$\rho_{X,Y} = -0.80$，这表明[DET]令牌彼此接近时（*i.e*即具有高余弦相似度），通常也会产生位置邻近的预测结果（*i.e.*即ℓ_2距离较短，给定$\rho_{X,Y} < 0$）。

我们还对X =标记对[DET]的余弦相似度与Y =分类器输出特征的相应余弦相似度之间的关系进行了定量研究。结果为$\rho_{X,Y} = -0.07$，非常接近于0。这意味着这两个变量之间不存在强线性相关性。

分离检测令牌。为了进一步理解[DET]令牌所扮演的角色，我们研究了在训练过程中分离YOLOS的[DET]令牌所产生的影响，*i.e.*，即不优化那一百个随机初始化的[DET]令牌参数。如表7所示，分离[DET]令牌对AP的影响较小。这些结果表明，[DET]令牌主要作为[PATCH]令牌的信息载体。类似现象在Fang等人的研究[22]中也有观察到。

Model	[DET] Tokens Config	AP
YOLOS-Ti	Rand. Init. & Learnable	28.7
	Rand. Init. & **Detached**	28.3
YOLOS-S	Rand. Init. & Learnable	36.1
	Rand. Init. & **Detached**	36.4

表7：训练期间分离YOLOS的[DET]标记的影响。

4 相关工作

视觉Transformer在目标检测中的应用。结合卷积神经网络(CNN)与自注意力机制以提升目标检测性能的研究备受关注[4,9,31,63]，而近期工作更倾向于用CNN（或CNN设计）增强Transformer架构。Beal等人[6]提出使用预训练ViT作为Faster R-CNN[50]目标检测器的特征提取器，虽取得成效，但未对CNN架构、区域池化操作[23,25,27]以及密集锚框[50]、非极大值抑制等人工设计组件进行消融研究。受现代CNN架构启发，部分研究[39,59,62,65]将金字塔特征层次结构与局部性引入视觉Transformer设计，显著提升了包括目标检测在内的密集预测任务性能。然而这些架构以性能为导向，未能体现直接继承自Vaswani等人[58]的标准（vanilla）视觉Transformer[21]特性。另一系列工作——DETR家族[10,72]采用随机初始化的Transformer对CNN特征进行编解码来实现目标检测，但未能揭示预训练Transformer的可迁移性。

UP-DETR [16] is probably the first to study the effects of unsupervised pre-training in the DETR framework, which proposes an "object detection oriented" unsupervised pre-training task tailored for Transformer encoder & decoder in DETR. In this paper, we argue for the characteristics of a pre-trained vanilla ViT in object detection, which is rare in the existing literature.

Pre-training and Fine-tuning of Transformer. The textbook-style usage of Transformer [58] follows a "pre-training & fine-tuning" paradigm. In NLP, Transformer-based models are often pre-trained on large corpora and then fine-tuned for different tasks at hand [18, 44]. In computer vision, Dosovitskiy et al. [21] apply Transformer to image recognition at scale using modern vision transfer learning recipe [33]. They show that a standard Transformer encoder architecture is able to attain excellent results on mid-sized or small image recognition benchmarks (*e.g*, ImageNet-1k [51], CIFAR-10/100 [34], *etc.*) when pre-trained at sufficient scale (*e.g*, JFT-300M [55], ImageNet-21k [17]). Touvron et al. [57] achieves competitive Top-1 accuracy by training Transformer on ImageNet-1k only, and is also capable of transferring to smaller datasets [34, 42, 43]. However, existing transfer learning literature of Transformer arrest in image-level recognition and does not touch more complex tasks in vision such as object detection, which is also widely used to benchmark CNNs transferability.

Our work aims to bridge this gap. We study the performance and properties of ViT on the challenging COCO object detection benchmark [36] when pre-trained on the mid-sized ImageNet-1k dataset [51] using different strategies.

5 Discussion

Over recent years, the landscape of computer vision has been drastically transformed by Transformer, especially for recognition tasks [10, 21, 39, 57, 59]. Inspired by modern CNN design, some recent works [39, 59, 62, 65] introduce the pyramidal feature hierarchy as well as locality to vanilla ViT [21], which largely boost the performance in dense recognition tasks including object detection.

We believe there is nothing wrong to make performance-oriented architectural designs for Transformer in vision, as choosing the right inductive biases and priors for target tasks is crucial for model design. However, we are more interested in designing and applying Transformer in vision following the spirit of NLP, *i.e.*, pre-train the *task-agnostic* vanilla Vision Transformer for general visual representation learning first, and then fine-tune or adapt the model on specific target downstream tasks *efficiently*. Current state-of-the-art language models pre-trained on massive amounts of corpora are able to perform few-shot or even zero-shot learning, adapting to new scenarios with few or no labeled data [8, 38, 45, 46]. Meanwhile, prevalent pre-trained computer vision models, including various Vision Transformer variants, still need a lot of supervision to transfer to downstream tasks.

We hope the introduction of Transformer can not only unify NLP and CV in terms of the architecture, but also in terms of the methodology. The proposed YOLOS is able to turn a pre-trained ViT into an object detector with the fewest possible *modifications*, but our ultimate goal is to adapt a pre-trained model to downstream vision tasks with the fewest possible *costs*. YOLOS still needs 150 epochs transfer learning to adapt a pre-trained ViT to perform object detection, and the detection results are far from saturating, indicating the pre-trained representation still has large room for improvement. We encourage the vision community to focus more on the general visual representation learning for the *task-agnostic* vanilla Transformer instead of the *task-oriented* architectural design of ViT. We hope one day, in computer vision, a universal pre-trained visual representation can be easily adapted to various understanding as well as generation tasks with the fewest possible *costs*.

6 Conclusion

In this paper, we have explored the transferability of the vanilla ViT pre-trained on mid-sized ImageNet-1k dataset to the more challenging COCO object detection benchmark. We demonstrate that 2D object detection can be accomplished in a pure sequence-to-sequence manner with minimal additional inductive biases. The performance on COCO is promising, and these preliminary results are meaningful, suggesting the versatility and generality of Transformer to various downstream tasks.

UP-DETR [16] 可能是首个研究DETR框架中无监督预训练效果的工作，它提出了一种专为DETR中Transformer编码器与解码器设计的“面向目标检测”的无监督预训练任务。本文中，我们探讨了预训练原始ViT在目标检测中的特性，这一研究在现有文献中较为罕见。

Transformer的预训练与微调。教科书式的Transformer应用[58]遵循“预训练&微调”范式。在自然语言处理领域，基于Transformer的模型通常先在大规模语料库上进行预训练，随后针对不同任务进行微调[18,44]。计算机视觉领域，Dosovitskiy等人[21]采用现代视觉迁移学习方案[33]，将Transformer应用于大规模图像识别。研究表明，当预训练规模足够大时（*e.g*，如JFT-300M[55]、ImageNet-21*k*[17]），标准Transformer编码器架构能在中型或小型图像识别基准（*e.g*、ImageNet-1*k*[51]、CIFAR-10/100[34]、*etc.*）上取得优异结果。Touvron团队[57]仅通过ImageNet-1*k*训练Transformer就获得了具有竞争力的Top-1准确率，并证明其可迁移至更小规模数据集[34,42,43]。然而现有Transformer迁移学习研究多停留在图像级识别任务，尚未涉足目标检测等更复杂的视觉任务——这些任务同样被广泛用于评估CNN的迁移能力。

我们的工作旨在弥合这一差距。我们研究了在中等规模的ImageNet-1*k*数据集[51]上采用不同预训练策略时，ViT在具有挑战性的COCO目标检测基准[36]上的表现与特性。

5 讨论

近年来，Transformer彻底改变了计算机视觉的格局，特别是在识别任务方面[10, 21, 39, 57, 59]。受现代CNN设计的启发，一些最新研究[39, 59, 62, 65]为原始ViT[21]引入了金字塔特征层次结构及局部性概念，这极大提升了包括目标检测在内的密集识别任务性能。

我们认为，为视觉领域的Transformer进行以性能为导向的架构设计并无不妥，因为为目标任务选择合适的归纳偏置和先验知识对模型设计至关重要。然而，我们更倾向于遵循自然语言处理的精神*i.e.*，首先预训练*task-agnostic*标准视觉Transformer以进行通用视觉表示学习，随后在特定下游任务*efficiently*上对模型进行微调或适配。当前最先进的语言模型通过海量语料库预训练后，能够实现少样本甚至零样本学习，仅需少量或无需标注数据即可适应新场景[8, 38, 45, 46]。而主流的预训练计算机视觉模型，包括各类Vision Transformer变体，在迁移至下游任务时仍需要大量监督信息。

我们希望Transformer的引入不仅能统一NLP与CV的架构，还能在方法论层面实现统一。提出的YOLOS能够以最少的*modifications*将预训练ViT转化为目标检测器，但我们的终极目标是以最少的*costs*使预训练模型适配下游视觉任务。YOLOS仍需150个epoch的迁移学习来调整预训练ViT以执行目标检测，且检测结果远未饱和，这表明预训练表征仍有巨大改进空间。我们呼吁视觉社区更关注*task-agnostic*原生Transformer的通用视觉表征学习，而非*task-oriented*ViT的结构设计。期待有朝一日，在计算机视觉领域，一个通用的预训练视觉表征能以最少的*costs*轻松适配各类理解与生成任务。

6 结论

本文探讨了基于中等规模ImageNet-1*k*数据集预训练的原始ViT模型向更具挑战性的COCO目标检测基准的迁移能力。我们证明，二维目标检测可以纯粹以序列到序列的方式完成，仅需引入极少的额外归纳偏置。在COCO数据集上的表现令人鼓舞，这些初步成果具有重要意义，表明Transformer架构对于各类下游任务具有广泛的适用性与通用性。

Acknowledgment

This work is in part supported by NSFC (No. 61876212, No. 61733007, and No. 61773176) and the Zhejiang Laboratory under Grant 2019NB0AB02. We thank Zhuowen Tu for valuable suggestions.

References

[1] Edward H Adelson, Charles H Anderson, James R Bergen, Peter J Burt, and Joan M Ogden. Pyramid methods in image processing. *RCA engineer*, 1984.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[3] Alexei Baevski and Michael Auli. Adaptive input representations for neural language modeling. *arXiv preprint arXiv:1809.10853*, 2018.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2015.

[5] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[6] Josh Beal, Eric Kim, Eric Tzeng, Dong Huk Park, Andrew Zhai, and Dmitry Kislyuk. Toward transformer-based object detection. *arXiv preprint arXiv:2012.09958*, 2020.

[7] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

[8] 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.

[9] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *ICCV*, 2019.

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

[11] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021.

[12] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, 2020.

[13] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021.

[14] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPRW*, 2020.

[15] Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Bichen Wu, Zijian He, Zhen Wei, Kan Chen, Yuandong Tian, Matthew Yu, Peter Vajda, et al. Fbnetv3: Joint architecture-recipe search using neural acquisition function. *arXiv preprint arXiv:2006.02049*, 2020.

[16] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. In *CVPR*, 2021.

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

致谢

本研究部分得到国家自然科学基金（编号：61876212、61733007、61773176）及之江实验室项目（2019NB0AB02）资助。感谢朱文图（Zhuowen Tu）提出的宝贵建议。

参考文献

[1] 爱德华·H·阿德尔森、查尔斯·H·安德森、詹姆斯·R·伯根、彼得·J·伯特与琼·M·奥格登。图像处理中的金字塔方法。{v*}，1984年。

[2] Jimmy Lei Ba、Jamie Ryan Kiros 和 Geoffrey E Hinton。层归一化。*arXiv preprint arXiv:1607.06450*，2016年。

[3] Alexei Baevski 与 Michael Auli。面向神经语言建模的自适应输入表示。*arXiv preprint arXiv:1809.10853*，2018年。

[4] 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*, 2015。

[5] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner等。关系归纳偏差、深度学习与图网络。*arXiv preprint arXiv:1806.01261*，2018年。

[6] 乔希·比尔、埃里克·金、埃里克·曾、朴东赫、安德鲁·翟与德米特里·基斯柳克。迈向基于Transformer的目标检测。*arXiv preprint arXiv:2012.09958*，2020年。

[7] Samuel R Bowman, Gabor Angeli, Christopher Potts, 与 Christopher D Manning。一个用于学习自然语言推理的大规模标注语料库。*arXiv preprint arXiv:1508.05326*，2015年。

[8] 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。{v*}，2020年。

[9] 曹越，徐佳瑞，林史蒂芬，魏芳芸，胡涵。GCNet：非局部网络与挤压激励网络的融合与超越。发表于*ICCV*，2019年。

[10] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, 和 Sergey Zagoruyko。基于transformer的端到端目标检测。发表于*ECCV*，2020年。

[11] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, 与 Armand Joulin。自监督视觉Transformer中的新兴特性。*arXiv preprint arXiv:2104.14294*，2021年。 [12] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, 及 Ilya Sutskever。基于像素的生成式预训练。收录于*ICML*，2020年。 [13] 陈新雷，谢赛宁，何凯明。自监督视觉Transformer训练的实证研究。*arXiv preprint arXiv:2104.02057*，2021年。 [14] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, 与 Quoc V Le。Randaugment：缩减搜索空间的实用自动化数据增强方法。收录于*CVPRW*，2020年。 [15] 戴晓亮，Alvin Wan，张培昭，吴必灿，何子健，魏震，陈侃，田渊栋，Matthew Yu，Peter Vajda等。Fbnetv3：利用神经采集函数联合架构-配方搜索。*arXiv preprint arXiv:2006.02049*，2020年。 [16] 戴志刚，蔡博伦，林雨耕，陈俊颖。UP-DETR：基于Transformer的目标检测无监督预训练。收录于*CVPR*，2021年。 [17] 邓嘉，董伟，Richard Socher，李立佳，李凯，与李飞飞。ImageNet：一个大规模分层图像数据库。收录于*CVPR*，2009年。

[18] 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.

[19] William B Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *IWP*, 2005.

[20] Piotr Dollár, Mannat Singh, and Ross Girshick. Fast and accurate model scaling. *arXiv preprint arXiv:2103.06877*, 2021.

[21] 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. *arXiv preprint arXiv:2010.11929*, 2020.

[22] Jiemin Fang, Lingxi Xie, Xinggang Wang, Xiaopeng Zhang, Wenyu Liu, and Qi Tian. Msg-transformer: Exchanging local spatial information by manipulating messenger tokens. *arXiv preprint arXiv:2105.15168*, 2021.

[23] Ross Girshick. Fast r-cnn. In *ICCV*, 2015.

[24] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *TPAMI*, 2015.

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

[27] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. In *ICCV*, 2017.

[28] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *ICCV*, 2019.

[29] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[30] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[31] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *CVPR*, 2018.

[32] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *ECCV*, 2016.

[33] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. *arXiv preprint arXiv:1912.11370*, 2019.

[34] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[35] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 2012.

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

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

[38] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021.

[18] 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。

[19] William B Dolan 与 Chris Brockett。自动构建句子复述语料库。载于*IWP*，2005年。

[20] Piotr Dollár、Mannat Singh 和 Ross Girshick。快速且准确的模型缩放。*arXiv preprint arXiv:2103.06877*，2021年。[21] 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。*arXiv preprint arXiv:2010.11929*，2020年。

[22] 方杰民，谢凌曦，王兴刚，张晓鹏，刘文予，田奇。Msg-Transformer：通过操控信使令牌交换局部空间信息。*arXiv preprint arXiv:2105.15168*，2021。

[23] Ross Girshick. Fast R-CNN. 载于*ICCV*, 2015.

[24] 伊恩·古德费洛、约书亚·本吉奥与亚伦·库维尔。{v*}。麻省理工学院出版社，2016年。

[25] 何恺明、张翔宇、任少卿、孙剑。深度卷积网络中用于视觉识别的空间金字塔池化方法。*TPAMI*，2015年。

[26] 何恺明、张翔宇、任少卿、孙剑。深度残差学习在图像识别中的应用。发表于*CVPR*，2016年。

[27] 何恺明、Georgia Gkioxari、Piotr Dollár与Ross B. Girshick。Mask R-CNN。发表于*ICCV*，2017年。 [28] 何恺明、Ross Girshick与Piotr Dollár。重新思考ImageNet预训练。发表于*ICCV*，2019年。

[29] Dan Hendrycks 与 Kevin Gimpel。高斯误差线性单元（GELUs）。*arXiv preprint arXiv:1606.08415*，2016年。

[30] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, 和 Hartwig Adam。MobileNets：面向移动视觉应用的高效卷积神经网络。*arXiv preprint arXiv:1704.04861*，2017年。

[31] 韩虎、顾家元、张政、戴继峰和魏亦忱。用于目标检测的关系网络。收录于*CVPR*，2018年。

[32] 高黄、孙宇、庄子刘、Daniel Sedra 和 Kilian Q Weinberger。随机深度深度网络。载于*ECCV*，2016年。

[33] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, 和 Neil Houlsby。大规模迁移学习（BiT）：通用视觉表示学习。*arXiv preprint arXiv:1912.11370*，2019年。

[34] Alex Krizhevsky、Geoffrey Hinton等。从小图像中学习多层特征。2009年。

[35] Alex Krizhevsky、Ilya Sutskever 和 Geoffrey E Hinton。使用深度卷积神经网络进行ImageNet分类。*NeurIPS*，2012年。 [36] Tsung-Yi Lin、Michael Maire、Serge Belongie、James Hays、Pietro Perona、Deva Ramanan、Piotr Dollár 和 C Lawrence Zitnick。Microsoft COCO：上下文中的常见物体。载于*ECCV*，2014年。 [37] Tsung-Yi Lin、Piotr Dollár、Ross Girshick、Kaiming He、Bharath Hariharan 和 Serge Belongie。用于目标检测的特征金字塔网络。载于*CVPR*，2017年。 [38] Pengfei Liu、Weizhe Yuan、Jinlan Fu、Zhengbao Jiang、Hiroaki Hayashi 和 Graham Neubig。预训练、提示与预测：自然语言处理中提示方法的系统综述。*arXiv preprint arXiv:2107.13586*，2021年。

[39] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

[40] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[41] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[42] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008.

[43] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *CVPR*, 2012.

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

[45] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

[46] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[47] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *CVPR*, 2020.

[48] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[49] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[50] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015.

[51] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015.

[52] Erik F Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*, 2003.

[53] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[54] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 2014.

[55] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 2017.

[56] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019.

[57] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[39] 刘泽、林宇彤、曹越、胡涵、魏亦轩、张政、Stephen Lin 和郭柏宁。Swin Transformer：基于移位窗口的分层视觉Transformer。*arXiv preprint arXiv:2103.14030*，2021年。

[40] Ilya Loshchilov 和 Frank Hutter。解耦权重衰减正则化。*arXiv preprint arXiv:1711.05101*，2017年。

[41] Vinod Nair 和 Geoffrey E Hinton。修正线性单元改进受限玻尔兹曼机。发表于*ICML*，2010年。 [42] Maria-Elena Nilsback 和 Andrew Zisserman。大规模花卉类别的自动分类。发表于*ICVGIP*，2008年。 [43] Omkar M. Parkhi、Andrea Vedaldi、Andrew Zisserman 和 C. V. Jawahar。猫与狗。发表于*CVPR*，2012年。 [44] Alec Radford、Karthik Narasimhan、Tim Salimans 和 Ilya Sutskever。通过生成式预训练提升语言理解能力。2018年。

[45] 亚历克·拉德福德、杰弗里·吴、雷温·柴尔德、戴维·卢安、达里奥·阿莫代伊和伊利亚·苏茨克弗。语言模型是无监督多任务学习者。*OpenAI blog*，2019年。

[46] Alec Radford、Jong Wook Kim、Chris Hallacy、Aditya Ramesh、Gabriel Goh、Sandhini Agarwal、Girish Sastry、Amanda Askell、Pamela Mishkin、Jack Clark等。从自然语言监督中学习可迁移的视觉模型。*arXiv preprint arXiv:2103.00020*，2021年。

[47] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, 何恺明, 和 Piotr Dollár。设计网络的设计空间。发表于*CVPR*，2020年。

[48] Pranav Rajpurkar、Jian Zhang、Konstantin Lopyrev和Percy Liang。SQuAD：面向机器理解文本的10万+个问题。*arXiv preprint arXiv:1606.05250*，2016年。

[49] Joseph Redmon 与 Ali Farhadi。Yolov3：渐进式改进。*arXiv preprint arXiv:1804.02767*，2018年。

[50] 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。*arXiv preprint arXiv:1506.01497*，2015年。

[51] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein等。ImageNet大规模视觉识别挑战赛。*IJCV*，2015年。

[52] Erik F Sang 和 Fien De Meulder。CoNLL-2003共享任务简介：语言无关的命名实体识别。*arXiv preprint cs/0306050*，2003年。

[53] Karen Simonyan 与 Andrew Zisserman。用于大规模图像识别的极深度卷积网络。*arXiv preprint arXiv:1409.1556*，2014年。

[54] 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。*JMLR*, 2014。

[55] 陈孙、阿比纳夫·施里瓦斯塔瓦、索拉布·辛格与阿比纳夫·古普塔。《重探大数据在深度学习时代的非凡效力》。发表于*ICCV*，2017年。[56] 谭明星与黎国。《EfficientNet：重新思考卷积神经网络的模型缩放方法》。发表于*ICML*，2019年。

[57] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, 和 Hervé Jégou。通过注意力机制训练数据高效的图像变换器及蒸馏方法。*arXiv preprint arXiv:2012.12877*，2020年。

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, 和 Illia Polosukhin。注意力就是你需要的一切。*arXiv preprint arXiv:1706.03762*，2017年。

[59] Ashish Vaswani, Prajit Ramachandran, Aravind Srinivas, Niki Parmar, Blake Hechtman, and Jonathon Shlens. Scaling local self-attention for parameter efficient visual backbones. *arXiv preprint arXiv:2103.12731*, 2021.

[60] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Scaled-yolov4: Scaling cross stage partial network. *arXiv preprint arXiv:2011.08036*, 2020.

[61] Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. Learning deep transformer models for machine translation. *arXiv preprint arXiv:1906.01787*, 2019.

[62] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*, 2021.

[63] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

[64] Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

[65] Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. *arXiv preprint arXiv:2104.06399*, 2021.

[66] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019.

[67] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[68] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[69] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI*, 2020.

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

[71] Benjin Zhu, Jianfeng Wang, Zhengkai Jiang, Fuhang Zong, Songtao Liu, Zeming Li, and Jian Sun. Autoassign: Differentiable label assignment for dense object detection. *arXiv preprint arXiv:2007.03496*, 2020.

[72] 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.

[59] Ashish Vaswani、Prajit Ramachandran、Aravind Srinivas、Niki Parmar、Blake Hechtman 和 Jonathon Shlens。为参数高效的视觉骨干网络扩展局部自注意力机制。*arXiv preprint arXiv:2103.12731*，2021年。

[60] 王建尧、Alexey Bochkovskiy与廖弘源。Scaled-YOLOv4：跨阶段局部网络的缩放。*arXiv preprint arXiv:2011.08036*，2020年。

[61] 王强，李北，肖桐，朱靖波，李长亮，Derek F Wong，赵丽霞。学习深度Transformer模型用于机器翻译。*arXiv preprint arXiv:1906.01787*，2019年。[62] 王文海，谢恩泽，李翔，范登平，宋凯涛，梁定，卢同，罗平，邵岭。金字塔视觉Transformer：一种无需卷积的密集预测通用骨干网络。*arXiv preprint arXiv:2102.12122*，2021年。

[63] 王小龙、Ross Girshick、Abhinav Gupta与何恺明。非局部神经网络。发表于*CVPR*，2018年。

[64] Ross Wightman. PyTorch图像模型. https://github.com/rwightman/pytorch-image-models, 2019.

[65] 徐伟健、徐一凡、Tyler Chang与涂卓文。共尺度卷积注意力图像变换器。*arXiv preprint arXiv:2104.06399*，2021年。

[66] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, 和 Youngjoon Yoo。CutMix：一种训练具有可定位特征的强分类器的正则化策略。发表于*ICCV*，2019年。

[67] Sergey Zagoruyko 和 Nikos Komodakis。宽残差网络。*arXiv preprint arXiv:1605.07146*，2016年。[68] Hongyi Zhang、Moustapha Cisse、Yann N Dauphin 和 David Lopez-Paz。mixup：超越经验风险最小化。*arXiv preprint arXiv:1710.09412*，2017年。

[69] 钟准，郑亮，康国良，李少子，杨毅。随机擦除数据增强。载于*AAAI*，2020年。

[70] 周行易、王德全与Philipp Krähenbühl。物体即点。*arXiv preprint arXiv:1904.07850*，2019年。

[71] 朱本金，王建峰，蒋正凯，宗福航，刘松涛，李泽明，孙剑。Autoassign：密集目标检测中的可微分标签分配方法。*arXiv preprint arXiv:2007.03496*，2020年。

[72] 朱熙舟，苏伟杰，卢乐为，李斌，王晓刚，戴继峰。可变形DETR：端到端目标检测中的可变形Transformer。*arXiv preprint arXiv:2010.04159*，2020年。

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Sec. 3.2, Sec. 3.3 and Sec. 3.4.
   (b) Did you describe the limitations of your work? [Yes] See Sec. 3.2, Sec. 3.3 and Sec. 3.4.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Sec. 3.2 and Tab. 2 for the total theoretical computations analysis.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] .
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We include them in the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Sec. 3.1.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Appendix.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Sec. 3.1 and Sec. 3.2.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] In the supplementary material.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] In the supplementary material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

检查清单

1. 对所有作者...

   (a) 摘要和引言中提出的主要主张是否准确反映了论文的贡献与范围？[是] 参见第3.2节、第3.3节及第3.4节。 (b) 是否描述了工作的局限性？[是] 参见第3.2节、第3.3节及第3.4节。 (c) 是否讨论了工作可能带来的负面社会影响？[是] 关于理论计算的总量分析，参见第3.2节及表2。 (d) 是否阅读了伦理审查指南并确保论文符合其要求？[是]

2. 若包含理论结果...

   (a) 你是否陈述了所有理论结果的完整假设集？[不适用] (b) 你是否包含了所有理论结果的完整证明？[不适用]

3. 如果你进行了实验...

   (a) 是否包含了重现主要实验结果所需的代码、数据和说明（无论是在补充材料中还是以URL形式提供）？[是] 我们已将其包含在补充材料中。 (b) 是否详细说明了所有训练细节（例如数据划分、超参数及其选择依据）？[是] 详见第3.1节。 (c) 是否报告了误差范围（例如通过多次实验运行得到的随机种子相关结果）？[是] 详见附录。 (d) 是否说明了使用的总计算量及资源类型（如GPU型号、内部集群或云服务提供商）？[是] 详见第3.1节和第3.2节。

4. 如果您正在使用现有资源（如代码、数据、模型）或整理/发布新资源...

   (a) 如果您的作品使用了现有资源，是否引用了创作者？[是] (b) 是否提及了资源的许可协议？[是] 在补充材料中。 (c) 是否在补充材料或通过URL提供了任何新资源？[是] 在补充材料中。 (d) 是否讨论了从数据提供者/整理者处获取同意的情况及方式？[不适用] (e) 是否讨论了您使用/整理的数据是否包含个人身份信息或冒犯性内容？[不适用]

5. 如果您使用了众包或与人类受试者进行了研究...

   (a) 是否包含了提供给参与者的完整说明文本及适用情况下的截图？[不适用] (b) 是否描述了任何潜在的参与者风险，并附上机构审查委员会（IRB）批准的链接（如适用）？[不适用] (c) 是否注明了支付给参与者的估计小时工资及参与者补偿的总支出？[不适用]