SimPLR: A Simple and Plain Transformer for Object Detection and Segmentation

Duy-Kien Nguyen Martin R. Oswald Cees G. M. Snoek
Atlas Lab - University of Amsterdam
{d.k.nguyen, m.r.oswald, cgmsnoek}@uva.nl

Abstract

The ability to detect objects in images at varying scales has played a pivotal role in the design of modern object detectors. Despite considerable progress in removing hand-crafted components and simplifying the architecture with transformers, multi-scale feature maps and/or pyramid design remain a key factor for their empirical success. In this paper, we show that this reliance on either feature pyramids or an hierarchical backbone is unnecessary and a transformer-based detector with scale-aware attention enables the plain detector 'SimPLR' whose backbone and detection head are both non-hierarchical and operate on single-scale features. The plain architecture allows SimPLR to effectively take advantages of self-supervised learning and scaling approaches with ViTs, yielding competitive performance compared to hierarchical and multi-scale counterparts. We demonstrate through our experiments that when scaling to larger ViT backbones, SimPLR indicates better performance than end-to-end segmentation models (Mask2Former) and plain-backbone detectors (ViTDet), while consistently being faster. The code will be released.

1. Introduction

After its astonishing achievements in natural language processing, the transformer [42] has quickly become the neural network architecture of choice in computer vision, as evidenced by recent success in image classification [13, 32], object detection [3, 35, 51] and segmentation [8, 43, 50]. Unlike natural language processing, where the same pre-trained network can be deployed for a wide range of downstream tasks with only minor modifications [2, 11], computer vision tasks such as object detection and segmentation require a different set of domain-specific knowledge to be incorporated into the network. Consequently, it is commonly accepted that a modern object detector contains two main components: a pre-trained backbone as the *general* feature extractor, and a *task-specific* head that conducts detection and segmentation tasks using domain knowledge. For transformer-based vision architectures, the question remains whether to add more inductive biases or to learn them from data.

Figure 1. **A plain detector is non-trivial.** Even with the use of a plain backbone ViT pre-trained using MAE [19], feature pyramids are still important for both convolution-based (ViTDet [26]) and transformer-based (BoxeR [35]) detectors. While removing multi-scale input from the encoder, PlainDETR [30] still relies on feature pyramids for its box proposal generation and lags behind in performance. In this paper, we demonstrate that the plain detector, SimPLR, with the proposed scale-aware attention yields competitive performance compared to multi-scale counterparts.

The spatial nature of image data lies at the core of computer vision. Besides learning long range feature dependencies, the ability of capturing local structure of neighboring pixels is critical for representing and understanding the image content. Building upon the successes of convolutional neural networks, a line of research biases the transformer architecture to be *multi-scale* and *hierarchical* when dealing with the image input, *i.e.*, Swin Transformer [32] and others [14, 20, 44]. The hierarchical design makes it easy to create multi-scale features for dense vision tasks and allows pre-trained transformers to be seamlessly integrated into a convolution-based detection head with a feature pyramid network [29], yielding impressive results in object detection and segmentation. However, the inductive biases in the architectural design make it benefit less from self-supervised learning and the scaling of model size [26].

An alternative direction pursues the idea of a simple transformer with "less inductive biases" and emphasizes learning

vision-specific knowledge directly from image data. Specifically, the Vision Transformer (ViT) [13] stands out as a *plain* architecture with a constant feature resolution, and acts as the feature extractor in plain-backbone detection. This is motivated by the success of ViTs scaling behaviours in visual recognition [1, 9, 19]. In addition, the end-to-end detection framework proposed by Carion *et al*. [3] with a transformer-based detection head further removes many hand-designed components, like non-maximum suppression or intersection-over-union computation, that encodes the prior knowledge for object detection.

The plain design of ViTs, however, casts doubts about its ability to capture information of objects across multiple scales. While recent studies [13, 26] suggest that ViTs with global self-attention could potentially learn translation-equivariant and scale-equivariant properties during training, leading object detectors still require multi-scale feature maps and/or an hierarchical backbone. This observation holds true for both convolutional [18, 26, 37] and transformer-based detectors [8, 35, 51] (see Fig. 1). Unlike hierarchical backbones, the creation of feature pyramids conflicts with the original design philosophy of ViTs. Therefore, our goal is to pursue a plain detector whose backbone *and* detection head are both *single-scale* and *non-hierarchical*. This further simplifies the architecture for detection and segmentation in the pursuit of learning object representations from data.

In this paper, we introduce SimPLR, a plain detector for *both* end-to-end detection and segmentation frameworks [3, 8, 35, 51]. In particular, our detector extracts the single-scale feature map from a ViT backbone, which is then fed into the transformer encoder-decoder via a simple projection to make the prediction. To deal with objects of various sizes, we propose to incorporate scale information into the attention mechanism, resulting in an *adaptive-scale* attention. The proposed attention mechanism learns to capture adaptive scale distribution from training data. This eliminates the need for multi-scale feature maps from the ViT backbone, yielding a simple and efficient detector.

The proposed detector, SimPLR, turns out to be an effective solution for the plain detection and segmentation. We find that multi-scale feature maps are not necessary and the scale-aware attention mechanism adequately captures objects at various sizes from output features of a ViT backbone. Despite the plain architecture, our detector (SimPLR) shows competitive performance compared to the strong hierarchical-backbone or multi-scale detectors (*e.g*., ViTDet [26] and Mask2Former [8]), while being consistently faster. Moreover, the effectiveness of our detector is observed not only in object detection but also in instance and panoptic segmentation. Interestingly, the efficient design allows SimPLR to take advantages of the significant progress in self-supervised learning and scaling with ViTs (*e.g*., with MAE [19] and BEiT [36]), indicating plain detectors to be a promising direction for dense prediction tasks.

2. Related Work

Backbones for object detection. Inspired by R-CNN [16], modern object detection methods utilize a task-specific head on top of a pre-trained backbone. Initially, object detectors [17, 21, 39, 46] were dominated by convolutional neural network (CNN) backbones [24] pre-trained on ImageNet [10]. With the success of the transformer in learning from large-scale text data [2, 11], many studies have explored the transformer for computer vision [4, 13, 32]. Most recently, the Vision Transformer (ViT) [13] with a simple design demonstrated the capability of learning meaningful representation for visual recognition. By removing the need for labels, methods with self-supervised learning have emerged as an even more powerful solution for pre-training general vision representations [5, 19]. We show through experiments that SimPLR can take advantages of the significant progress in representation learning and scaling of ViTs.

End-to-end detection and segmentation. The end-to-end framework for object detection proposed in DETR [3] aims to remove the need for many hand-crafted modules. This is made possible by adopting Transformer as the detection head to directly give the prediction. Follow-up works [12, 35, 43] extended the transformer-based head in end-to-end frameworks for instance segmentation, and panoptic segmentation. This inspired MaskFormer [7] and K-Net [50] to unify segmentation tasks with a class-agnostic mask prediction. Pointing out that MaskFormer and K-Net lag behind specialized architectures, Cheng *et al*. [8] introduce Mask2Former, reaching strong performance on segmentation tasks. Yu *et al*. [47] replace self-attention with k-means clustering, boosting the effectiveness of the network. Another direction is to improve the object query in the decoder via a denoising process [25, 49]. While simplifying the detection and segmentation framework, these architectures still require an hierarchical backbone along with feature pyramids. The use of feature pyramids increases the sequence length of the input to the transformer-based detection head, making the detector less efficient. In this work, we enable a plain detector by removing hierarchical and multi-scale constraints.

Plain detectors. Following the goal of less inductive biases in the architecture, recent studies focus on the non-hierarchical and single-scale detector. Motivated by the success of ViT, a line of research considers plain-backbone detectors that replace the hierarchical backbone with a ViT. Initially, Chen *et al*. [6] present UViT as a plain detector that contains a ViT backbone and a single-scale convolutional detection head. Since the backbone architecture is modified *during pre-training* to adopt the progressive window attention, UViT is unable to take the advantages of existing pre-training approaches with ViTs. ViTDet [26] tackles this problem with simple adaptations of the ViT backbone

during fine-tuning. These simple modifications allow ViTDet to benefit directly from recent self-supervised learning with ViTs (*i.e.*, MAE [19]), resulting in strong results when scaling to larger models. Despite enabling plain-backbone detectors, feature pyramids are still an important factor in ViTDet to detect object at various scales.

Concurrent to our work, Lin *et al*. [30] introduce the transformer-based PlainDETR detector, which also removes the multi-scale input. However, it still relies on multi-scale features to generate the object proposals for its decoder. In the decoder, PlainDETR also uses hybrid matching [22] to strengthen its prediction, while our decoder preserves a simple design as in [35, 51]. We believe to be the first to remove the hierarchical and multi-scale constraints which appear in the backbone *and* the input of the transformer encoder for *both* detection and segmentation tasks. Our proposed scale-aware attention can further plug into current end-to-end frameworks without significant architectural changes.

3. Background

Our goal is to further simplify the detection and segmentation pipeline from [26, 35, 51], and to prove the effectiveness of the plain detector in object detection and segmentation tasks. To do so, we focus on the recent progress in end-to-end detection and segmentation. As a result, we utilize the sparse attention mechanism, Box-Attention in [35] and Deformable Attention in [51], as strong baselines due to its effectiveness in learning discriminative object representations while being lightweight in computation.

Multi-head Box-Attention. In box-attention, each query vector $q \in \mathbb{R}^d$ in the input feature map is assigned a reference window $r{=}[x, y, w, h]$, where x, y indicate the query coordinate and w, h are the size of the reference window both being normalized by the image size. The box-attention refines the reference window into a region of interest, r', as:

$$r' = F_{\text{scale}}\big(F_{\text{translate}}(r, q), q\big), \tag{1}$$

$$F_{\text{scale}}(r, q) = [x, y, w + \Delta_w, h + \Delta_h], \tag{2}$$

$$F_{\text{translate}}(r, q) = [x + \Delta_x, y + \Delta_y, w, h], \tag{3}$$

where F_{scale} and $F_{\text{translate}}$ are the scaling and translation transformations, $\Delta_x, \Delta_y, \Delta_w$ and Δ_h are the offsets regarding to the reference window r. A linear projection ($\mathbb{R}^d \rightarrow \mathbb{R}^4$) is applied on q to predict offset parameters (*i.e.*, $\Delta_x, \Delta_y, \Delta_w$ and Δ_h) w.r.t. the window size.

During the i-th attention head computation, a $2{\times}2$ feature grid is sampled from the corresponding region of interest r'_i, resulting in a set of value features $v_i \in \mathbb{R}^{2\times2\times d_h}$. The $2{\times}2$ attention scores are efficiently generated by computing a dot-product between $q \in \mathbb{R}^d$ and relative position embeddings ($k_i \in \mathbb{R}^{2\times2\times d}$) followed by a $\mathrm{softmax}$ function. The attended feature $\mathrm{head}_i \in \mathbb{R}^{d_h}$ is a weighted sum of $2{\times}2$ value features in v_i with the corresponding attention weights:

$$\alpha = \mathrm{softmax}(q^\top k_i), \tag{4}$$

$$\mathrm{head}_i = \mathrm{BoxAttention}(q, k_i, v_i) = \sum_{j=0}^{2\times2} \alpha_j v_{i_j}, \tag{5}$$

where $q \in \mathbb{R}^d$, $k_i \in \mathbb{R}^{2\times2\times d}$, $v_i \in \mathbb{R}^{2\times2\times d_h}$ are query, key and value vectors of box-attention, α_j is the j-th attention weight, and v_{i_j} is the j-th feature vector in the feature grid v_i. To better capture objects at different scales, the box-attention [35] takes t multi-scale feature maps, $\{e^j\}_{j=1}^t$, as its inputs in order to produce head_i.

Multi-head Deformable Attention. In deformable attention, each query vector $q \in \mathbb{R}^d$ is assigned a reference point $p = [x, y]$, where x, y indicate the query coordinate normalized by the image size. The deformable attention attends to points of interest around the query coordinate, p', as:

$$p' = F_{\text{deform}}(p, q) = [x + \Delta_x, y + \Delta_y], \tag{6}$$

where F_{deform} is the deformable function, Δ_x, Δ_y are the offets regarding to the reference point p. Similarly, a linear projection ($\mathbb{R}^d \rightarrow \mathbb{R}^2$) is applied on q to predict offset paramters w.r.t. the reference point.

During the i-th attention head computation, the deformable attention predicts a set of 4 points, resulting in value features $v_i \in \mathbb{R}^{4\times d_h}$. The 4 attention scores are generated by applying another linear projection, $g_i : \mathbb{R} \rightarrow \mathbb{R}^4$, on q followed by a $\mathrm{softmax}$ function. The attended feature $\mathrm{head}_i \in \mathbb{R}^{d_h}$ is a weighted sum of 4 value features in v_i with the corresponding attention weights:

$$\alpha = \mathrm{softmax}\big(g_i(q)\big), \tag{7}$$

$$\mathrm{head}_i = \mathrm{DeformableAttention}(q, v_i) = \sum_{j=0}^{4} \alpha_j v_{i_j}, \tag{8}$$

where $q \in \mathbb{R}^d$, $v_i \in \mathbb{R}^{2\times2\times d_h}$ are query, value vectors of deformable attention, α_j is the j-th attention weight, and v_{i_j} is the j-th feature vector in the value v_i. The deformable attention [51] also utilizes t multi-scale feature maps, $\{e^j\}_{j=1}^t$, as its inputs in order to produce head_i.

The sparse attention like box-attention and deformable attention lies at the core of recent end-to-end detection and segmentation models due to its ability of capturing object information with lower complexity. The effectiveness and efficiency of these attention mechanisms bring up the question: *Is multi-scale object information learnable within the detector which is non-hierarchical and single-scale?*

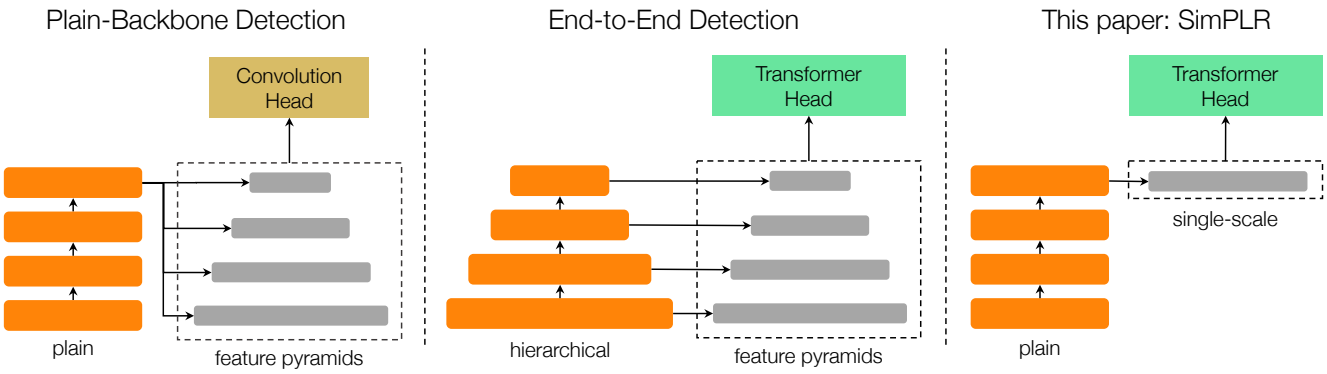


Figure 2. **Object detection architectures. Left:** The plain-backbone detector from [26] whose input (denoted in the dashed region) are multi-scale features. **Middle:** State-of-the-art end-to-end detectors [8, 35] utilize a hierarchical backbone (*i.e.*, Swin [32]) to create multi-scale inputs. **Right:** Our simple single-scale detector following the end-to-end framework. Where existing detectors require feature pyramids to be effective, we propose a plain detector, SimPLR, whose backbone and detection head are non-hierarchical and operate on a single-scale feature map. The plain detector, SimPLR, achieves on par or even better performance compared to hierarchical and/or multi-scale counterparts while being more efficient.

4. SimPLR: A Simple and Plain Detector

Multi-scale feature maps in a hierarchical backbone can be easily extracted from the pyramid structure [29, 31, 51]. When moving to a ViT backbone with a constant feature resolution, the creation of multi-scale feature maps requires complex backbone adaptations. Moreover, the benefits of multi-scale features in object detection frameworks using ViTs remain unclear. Recent studies on plain-backbone detection [6, 26] conjecture the high-dimensional ViT with self-attention and positional embeddings [42] is able to preserve important information for localizing objects[1]. From this conjecture, we hypothesize that a proper design of the transformer-based head will enable a plain detector.

Our proposed detector, SimPLR, is conceptually simple: a pre-trained ViT backbone to extract plain features from an image, which are then fed into a single-scale encoder-decoder to make the final prediction (See Fig. 2). Thus, SimPLR is a natural idea as it eliminates the non-trivial creation of feature pyramids from the ViT backbone. But the single-scale encoder-decoder requires an effective design to deal with objects at different scales. Next, we introduce the key elements of SimPLR, including its *scale-aware* attention that is the main factor for learning of adaptive object scales.

4.1. Scale-aware attention

The output features of the encoder should capture objects at different scales. Therefore, unlike the feature pyramids where each set of features encode a specific scale, predicting objects from a plain feature map requires its feature vectors to reason about dynamic scale information based on the image content. This can be addressed effectively by a multi-head attention mechanism that capture different scale information in each of its attention heads. In that case, global self-attention is a potential candidate because of its large receptive field and powerful representation. However, its computational complexity is quadratic w.r.t. the sequence length of the input, making the operation computationally expensive when dealing with high-resolution images. The self-attention also leads to worse performance and slow convergence in end-to-end detection [51]. This motivated us to develop a multi-head *scale-aware* attention mechanism based on sparse attention such as box-attention and deformable attention.

Scale-aware box-attention (SAB). The multi-head attention mechanism, proposed by [42], is a core operation in the transformer architectures for capturing diverse patterns given a query vector in the input features. In the multi-head box-attention [35], each feature vector is assigned a reference window which is then refined to locate a region-of-interest in each attention head via scaling and translation transformations. Theoretically, the scaling transformation should provide box-attention the capability to learn multi-scale regions-of-interest in multiple attention heads. However, we find in our experiments that the scaling function in box-attention only performs minor modifications regarding to its reference window. As a result, feature vectors learn to adapt to a specific scale of the reference window assigned to them. While this behaviour may not impact the multi-scale box-attention – which utilizes feature pyramids for detecting objects – it poses a big challenge in learning scale-equivariant features on a single-scale input.

To address this limitation, we propose two variants of multi-head *scale-aware* attention (*i.e.*, *fixed-scale* and

[1] ViT-B and larger (dim $\geq$ 768) can maintain information with a patch size of $16\times16\times3$ in the input image.

adaptive-scale) that integrate different scales into each attention head, allowing query vectors to choose the suitable scale information during training. Our proposed attention mechanism is simple: we assign reference windows of m different scales to attention heads of each query. We use m reference windows with size $w{=}h \in \{s \cdot 2^j\}_{j=0}^{m-1}$, where s is the size of the smallest window, and m is the number of scales. Surprisingly, our experiments show that the results are not sensitive to the size of the window, as long as *enough number of scales* are used.

i) *Fixed-Scale Attention.* Given reference windows of m scales, we distribute them to n attention heads in a round-robin manner. Thus, in multi-head fixed-scale attention, $\frac{n}{m}$ attention heads are allocated for each of the window scales. This uniform distribution of different scales enables fixed-scale attention to learn diverse information from local to more global context. The aggregation of n heads results in scale-aware features, that is suitable for predicting objects of different sizes.

ii) *Adaptive-Scale Attention.* Instead of uniformly assigning m scales to n attention heads, the adaptive-scale attention learns to allocate a scale distribution based on the context of the query vector. This comes from the motivation that the query vector belonging to a small object should use more attention heads for capturing fine-grained details rather than global context, and vice versa.

Given the query vector $q \in \mathbb{R}^d$ in the input feature map and m reference windows of different scales, $\{r^j\}_{j=0}^{m-1}$, the adaptive-scale attention predicts offsets of all reference windows, $\{\Delta_{x_j}, \Delta_{y_j}, \Delta_{w_j}, \Delta_{h_j}\}_{j=0}^{m-1}$, in each attention head. Besides, we apply a scale temperature to each set of offsets before the transformations:

$$F_{\text{scale}}(r_j, q) = [x, y, w + \Delta_w \cdot \frac{2^j}{\lambda}, h + \Delta_h \cdot \frac{2^j}{\lambda}], \quad (9)$$

$$F_{\text{translate}}(r_j, q) = [x + \Delta_x \cdot \frac{2^j}{\lambda}, y + \Delta_y \cdot \frac{2^j}{\lambda}, w, h], \quad (10)$$

where $\frac{2^j}{\lambda}$ is the scale temperature corresponding to r_j. The scale temperature allows the transformation functions to capture regions of interest corresponding to the scale of reference windows. It then samples feature grids from m regions of interest and generates attention scores for these feature grids followed by $\mathrm{softmax}$ normalization. This makes our attention mechanism to focus on feature grids of suitable scale. The adaptive-scale attention provides efficiency due to sparse sampling and strong flexibility to control scale distribution via its attention computation.

Scale-aware deformable attention (SAD). Here, we adopt deformable attention to capture information of m scales. Instead of 4 sampled points per query, we predict $4 \cdot m$ points around the query coordinate. Each set of 4 points is first initialized at the corners of square whose center is the query coordinate and size is increased by a factor of 2. Similar to adaptive-scale box-attention, we apply a scale temperature to each set of offsets before the deformable function, yielding adaptive-scale deformable attention. This encourages each set of points to attend to regions corresponding to its scale.

4.2. Network Architecture

SimPLR follows the end-to-end detection and segmentation framework in [35] with the two-stage design. Specifically, we use a plain ViT as the backbone with 14×14 windowed attention and four equally-distributed global attention blocks as in [26]. In the detection head, the SimPLR encoder receives input features via a projection of the last feature map from the ViT backbone. The object proposals are then generated using single-scale features from the encoder and top-scored features are initialized as object queries for the SimPLR decoder to predict bounding boxes and masks.

Formally, we apply a projection f to the last feature map of the pre-trained ViT backbone, resulting in the input feature map $e \in \mathbb{R}^{H_e \times W_e \times d}$ where H_e, W_e are the size of the feature map, and d is the hidden dimension of the detection head. In SimPLR, the projection f is simply a single convolution projection, that provides us a great flexibility to control the resolution and dimension of the input features e to the encoder. The projection allows SimPLR to decouple the feature scale and dimension between its backbone and detection head to further improve the efficiency. This practice is different from the creation of SimpleFPN in [26] where a different stack of multiple convolution layers is used for each feature scale (more details shown in Fig. A in the supplementary material). We show by experiments that this formulation is key for plain detection and segmentation while keeping our network efficient.

Plain backbone. SimPLR deploys ViT as its plain backbone for feature extraction. We show that SimPLR can take advantages of recent progress in self-supervised learning with ViTs. To be specific, SimPLR generalizes to ViT backbones initialized by MAE [19] and BEiTv2 [36]. The efficient design of SimPLR allows us to effectively scale to larger ViT backbones which recently show to be even more powerful in learning representations [9, 19, 48]. We also provide the comparison between different pre-training approaches in the supplementary material.

Adaption for panoptic segmentation. Panoptic segmentation proposed by [23] requires the network to segment both "thing" and "stuff". To enable the plain detector on panoptic segmentation, we make an adaptation in the mask prediction of SimPLR. Following [8], we predict segmentation masks of both types by computing the dot-product between object queries and a feature map. We provide a brief description on these modifications, the full implementation details are provided in the supplementary material.

In [8], the $\frac{1}{4}$ feature scale is extracted from the first stage of the Swin and combined with upscaled $\frac{1}{8}$ features from

the last layer of the encoder for the mask prediction. As the ViT and SimPLR encoder features are of lower resolution, we simply interpolate the last encoder layer to $\frac{1}{4}$ scale and add a single attention layer on top. This simple modification produces a high resolution feature map that is beneficial for learning fine-grained details. Masked instance-attention follows the dense grid sampling strategy (*i.e.*, 14×14 feature grid) of box-attention in the decoder [35], but differs in the computation of the attention scores to better capture objects of different shapes. Inspired by masked self-attention [8], we employ the masking to 14×14 attention scores of the feature grid based on the mask prediction scores in the previous decoder layer. By focusing better on foreground features, the decoder yields more discriminative features.

5. Experiments

Experimental setup. In this study, we evaluate our method on COCO [27], a commonly used dataset for object detection, instance segmentation, and panoptic segmentation tasks. By default, we use plain features with adaptive-scale box-attention as described in Sec. 4 due to its strong performance and ability to perform both detection and segmentation; and initialize the ViT backbone from MAE [19] pre-trained on ImageNet-1K without any labels. In both fixed-scale and adaptive-scale attention, we set the number of scale $m = 4$ and the window size $s = 32$. Unless specified, the hyper-parameters are the same as in [35].

For all experiments, our optimizer is AdamW [33] with a learning rate of 0.0001. The learning rate is linearly warmed up for the first 250 iterations and decayed at 0.9 and 0.95 fractions of the total number of training steps by a factor 10. ViT-B [13] is set as the backbone. The input image size is 1024×1024 with large-scale jitter [15] between a scale range of $[0.1, 2.0]$. Due to the limit of our computational resources, we report the ablation study using the standard $5\times$ schedule setting with a batch size of 16 as in [35]. In the main experiments, we use the finetuning recipe from [26].

A single-scale detector is non-trivial. In Fig. 1, we explore the use of single-scale input feature by simply projecting the last feature map ($\frac{1}{16}$ scale) of the ViT backbone to the detection head. We first study the effect of the scaling transformation in box-attention on feature pyramids as it could possibly generate regions-of-interest at different scales. More specifically, we compare the area between generated regions and the initial reference window of query vectors corresponding to object proposals in the last encoder layer. Surprisingly, the change of area after scaling has a mean of 31% and standard deviation 33% w.r.t. the original area of the reference window (*e.g.*, for a reference window of 32×32 pixels to capture regions of different scales, such as 64×64 or 16×16 pixels, the change of the area should be more than 75%). This suggests that the scaling function in box-attention prefers to capture regions of different aspect ratios rather than multi-scale information.

	FLOPs head	Train mem.	FPS	AP^b	AP^m
Feature pyramids					
DETR from [30]	310G	-	-	46.5	n/a
DeformableDETR from [30]	280G	-	-	52.1	n/a
DeformableDETR (our impl.)	280G	1.2×	12	54.6	n/a
BoxeR	280G	1.3×	12	55.4	47.7
ViTDet w/ Cascade head	710G	-	11	54.0	46.7
Plain detector					
PlainDETR†	-	-	12	53.8	n/a
SimPLR w/ SAD	180G	0.9×	15	54.3	n/a
SimPLR w/ SAB	180G	1×	15	55.4	47.6

† Multi-scale features are used to generate object proposals.

Table 1. **SimPLR is an effective plain detector.** All methods use ViT-B as backbone. Methods that take feature pyramids as input employ SimpleFPN with ViT from [26]. Our plain detector, SimPLR, shows competitive performance compared to multi-scale alternatives, while being more efficient in terms of FLOPs, training memory and faster during inference. Training memory is reported as relative to SimPLR (SAD: scale-aware deformable attention; SAB: scale-aware box-attention).

It can be seen in Fig. 1 that deploying feature pyramids brings a large improvement to different types of detectors (*i.e.*, ∼2 AP points for BoxeR and ∼3 AP points for ViTDet). This observation is consistent to the observation in DeformableDETR [51] where the improvement of ∼2 AP points comes from feature pyramids. While removing multi-scale input to the detection head, the PlainDETR [30] still depends on features pyramids to generate object proposals, their performance drops by ∼1 AP when using single-scale features for proposal generation.

SimPLR is an effective single-scale detector. In Tab. 1, we show the comparison between SimPLR and recent object detectors using the plain backbone ViT. We also implement a strong baseline of DeformableDETR [51] with SimpleFPN under our end-to-end framework for better comparison. We report ViTDet with the setting in [26]; and other methods using the $5\times$ schedule. The plain detector, SimPLR, with both scale-aware box-attention (SAB) and scale-aware deformable attention (SAD) removes the need for multi-scale adaptation of the ViT.

While SimPLR with scale-aware deformable attention lags behind its multi-scale counterpart from our implementation, we observe a much smaller gap compared to standard deformable attention on single-scale input (*e.g.*, 0.3 *vs.* 2 AP point). When equipped with scale-aware box-attention, SimPLR reaches similar performance as BoxeR and outperforms other multi-scale detectors in both detection and segmentation. In addition, the plain detector is more efficient than multi-scale counterparts. When moving to larger models with higher dimension, we find that multi-scale de-

attention	AP^b	AP^m
base	53.6	46.1
i) fixed-scale	55.0	47.2
ii) adaptive-scale	55.4	47.6

(a) **Scale-aware attention.**

s	AP^b	AP^m
base	53.6	46.1
16	55.1	47.4
32	55.4	47.6
64	55.1	47.4

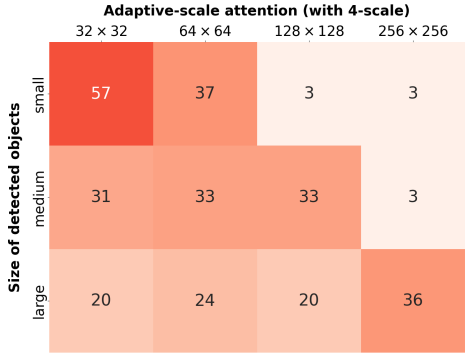
(b) **Window size.**

n	AP^b	AP^m
base	53.6	46.1
2	54.6	47.0
4	55.4	47.6
6	55.2	47.6

(c) **Number of window scales.**

scale	AP^b	AP^m
base	53.6	46.1
1/4	55.4	47.7
1/8	55.4	47.6
1/16	54.3	46.7

(d) **Scales of input features.**



(e) Visualization of scale distribution learnt in **multi-head adaptive-scale attention** of object proposals. Objects are classified into *small*, *medium*, and *large* based on their area.

Table 2. **Ablation of scale-aware attention** in SimPLR using a plain ViT backbone on COCO `val`. **Table (a-d):** Compared to the naïve baseline, which employs BoxeR and box-attention [35] with *single-scale* features, our plain detector, SimPLR, with scale-aware attention improves the performance consistently for all settings, default setting highlighted. **Figure e:** our adaptive-scale attention captures different scale distribution in its attention heads based on the context of query vectors. Specifically, queries of *small* objects tends to focus on reference windows of small scales (*i.e.*, mainly 32×32), while query vectors of *medium* and *large* objects distribute more attention computation into larger reference windows.

tectors like BoxeR require significant memory optimization and becomes challenging for our computational resources. We also compare with PlainDETR [30] which is a recent plain detection method. As discussed in Sec. 2, PlainDETR and our work approach the plain detector in different ways. PlainDETR aims at designing a strong decoder that compensates for single-scale input, while our goal is to learn scale equivariant features in backbone and encoder. Despite the different approaches, both PlainDETR and our work indicate that plain detection holds a great potential.

Ablation of scale-aware attention. Here, our baseline is the standard box-attention from [35] with single-scale feature input directly taken from the last feature of the ViT backbone (denoted as "base").

From Tab. 2a, we first conclude that *both* scale-aware attention strategies are substantially better than the naïve baseline, increasing AP by up to 1.8 points. We note that while fixed-scale attention distributes 25% of its attention heads into each of the window scales, adaptive-scale attention decides the scale distribution based on the query content. By choosing feature grids from different window scales adaptively, the adaptive-scale attention is able to learn a suitable scale distribution through training data, yielding better performance compared to fixed-scale attention. This is also verified in Fig. 2e where queries corresponding to *small* objects tend to pick reference windows of small sizes for its attention heads. Interestingly, queries corresponding to *medium* and *large* objects pick not only reference windows of their sizes, but also ones of smaller sizes. One of reasons may come from the fact that performing instance segmentation of larger objects still requires the network to faithfully preserve the per-pixel spatial details.

In Tab. 2b, we compare the performance of SimPLR across several sizes (s) of the reference window. They all improve over the baseline, while the choice of a specific base size makes only marginal differences. Our ablation reveals that the number of scales rather than the window size plays an important role to make our network more *scale-aware*. Indeed, in Tab. 2c, the use of 4 or more window scales shows improvement up to 0.8 AP over 2 window scales; and clearly outperforms the naïve baseline. Last, we show in Tab. 2d that the *decouple* between feature scale and dimension of the ViT backbone and the detection head features helps to boost the performance of our plain detector by $\sim$1 AP point, while keeping its efficiency (*i.e.*, both in terms of FLOPs and FPS). This practice makes scaling of SimPLR to larger ViT backbones more practical.

State-of-the-art comparison and scaling behavior. We show in Tabs. 3 and 4 that SimPLR indicates strong performance on object detection, instance segmentation and panoptic segmentation. To be specific, our plain detector combined with a ViT, pre-trained using MAE [19] or BEiTv2 [36], presents good scaling behavior. When moving to large and huge models, our method outperforms multi-scale counterparts including the recent end-to-end Mask2Former segmentation model [8]. Despite involving more advanced attention blocks designs, *i.e.*, shifted window attention in Swin [32] and pooling attention in MViT [14], detectors with hierarchical backbones benefit less from larger backbones. SimPLR is better than the plain-backbone detector, ViTDet, across all backbones in terms of both accuracy and inference speed.

Limitations. Our final goal is to simplify the detection

method	backbone	pre-train	Object Detection				Instance Segmentation				FPS
			AP^b	AP^b_S	AP^b_M	AP^b_L	AP^m	AP^m_S	AP^m_M	AP^m_L	
Base models											
Swin [32]	Swin-B	sup-22K	54.0	-	-	-	46.5	-	-	-	13
Mask2Former [8]	Swin-B	sup-22K			n/a		**48.1**	**27.8**	52.0	**71.1**	-
MViT [14]	MViT-B	sup-22K	55.6	-	-	-	**48.1**	-	-	-	11
BoxeR [35]	ViT-B	MAE	55.4	-	-	-	47.7	-	-	-	12
ViTDet [26]	ViT-B	MAE	54.0	36.5	57.8	69.1	46.7	27.2	49.6	64.9	11
UViT [6]	UViT-B	self-learning	53.9	-	-	-	46.1	-	-	-	12
PlainDETR [30]	ViT-B	MAE	53.8	35.9	57.0	68.9			n/a		12
SimPLR	ViT-B	MAE	55.6	**37.1**	59.2	71.5	48.0	27.5	51.5	67.8	**15**
SimPLR	ViT-B	BEiTv2	**55.7**	36.5	**60.2**	**72.3**	**48.1**	26.7	**52.7**	68.9	**15**
Large models											
Swin [32]	Swin-L	sup-22K	54.8	-	-	-	47.3	-	-	-	**10**
Mask2Former [8]	Swin-L	sup-22K			n/a		50.1	29.9	53.9	**72.1**	4
MViT [14]	MViT-L	sup-22K	55.7	-	-	-	48.3	-	-	-	6
ViTDet [26]	ViT-L	MAE	57.6	40.5	61.6	72.6	49.9	30.5	53.3	68.0	7
SimPLR	ViT-L	MAE	58.3	**42.2**	62.3	73.1	50.4	**32.0**	54.3	69.5	9
SimPLR	ViT-L	BEiTv2	**58.5**	40.1	**63.4**	**74.6**	**50.7**	30.2	**55.3**	70.8	9
Huge models											
MViT [26]	MViT-H	sup-22K	55.7	-	-	-	48.3	-	-	-	6
ViTDet [26]	ViT-H	MAE	58.7	**41.9**	63.0	73.9	50.9	**32.0**	54.3	68.9	5
SimPLR	ViT-H	MAE	**59.5**	41.8	**63.5**	**75.0**	**51.6**	31.7	**55.6**	**70.9**	**7**

Table 3. **State-of-the-art comparison and scaling behavior for object detection and instance segmentation.** We compare methods using feature pyramids (*top* row) *vs*. single-scale (*bottom* row) on COCO `val`. Backbones with MAE pre-trained on ImageNet-1K while others pre-trained on ImageNet-22K. Methods in gray color are with convolution-based detection head. (n/a: entry is not available). Models of larger sizes are grouped with *darker* orange color. SimPLR indicates good scaling behavior. With only single-scale features, SimPLR shows strong performance compared to multi-scale detectors including transformer-based detectors like Mask2Former, while being faster.

method	backbone	pre-train	Panoptic Segmentation			FPS
			PQ	PQ^{th}	PQ^{st}	
Base models						
MaskFormer [7]	Swin-B	sup-22K	51.8	56.9	44.1	-
Mask2Former [8]	Swin-B	sup-22K	56.4	62.4	**47.3**	-
SimPLR	ViT-B	BEiTv2	**56.5**	**62.6**	**47.3**	**13**
Large models						
Mask2Former [8]	Swin-L	sup-22K	57.8	64.2	48.1	4
SimPLR	ViT-L	BEiTv2	**58.5**	**65.1**	**48.6**	**8**

Table 4. **State-of-the-art comparison and scaling behavior for panoptic segmentation.** We compare between methods using feature pyramids (*top* row) *vs*. single-scale (*bottom* row) on COCO `val`. Models of larger sizes are with *darker* orange color. SimPLR shows better results when scaling to larger backbones, while being faster with single-scale input.

pipeline and to achieve competitive results at the same time. In Sec. 5, we find that the *adaptive-scale* attention mechanism that adaptively learns scale-aware information in its computation plays a key role for a plain detector. However, our adaptive-scale attention still encodes the knowledge of different scales. In the future, we hope that with the large-scale training data, a simpler design of the attention mechanism could also learn the scale equivariant property. Furthermore, SimPLR faces difficulties in detecting and segmenting large objects in the image. To overcome this limitation, we think that a design of attention computation which effectively combines both global and local information is necessary.

6. Conclusion

We presented SimPLR, a simple and plain object detector that eliminates the requirement for handcrafting multi-scale feature maps. Through our experiments, we demonstrated that a transformer-based detector, equipped with a scale-aware attention mechanism, can effectively learn scale-equivariant features through data. The efficient design of SimPLR allows it to take advantages of significant progress in scaling ViTs, reaching highly competitive performance on three tasks on COCO: object detection, instance segmentation, and panoptic segmentation. This finding suggests that many handcrafted designs for convolution neural network in computer vision could be removed when moving to transformer-based architecture. We hope this study could encourage future exploration in simplifying neural network architectures especially for dense vision tasks.

Acknowledgements

This work has been financially supported by TomTom, the University of Amsterdam and the allowance of Top consortia for Knowledge and Innovation (TKIs) from the Netherlands Ministry of Economic Affairs and Climate Policy.

A. Implementation Details

Masked Instance-Attention. The masked instance-attention follows the grid sampling strategy of the box-attention in [35], but differs in the computation of attention scores to better capture objects of different shapes. To be specific, the region of interest r'_i is divided into 4 bins of 2×2 grid, each of which contains a $\frac{m}{2} \times \frac{m}{2}$ grid features sampled using bilinear interpolation. Instead of assigning an attention weight to each feature vector, a linear projection ($\mathbb{R}^d \rightarrow \mathbb{R}^{2\times 2}$) is adopted to generate the 2×2 attention scores for 4 bins. The $\frac{m}{2} \times \frac{m}{2}$ feature vectors within the same bin share the same attention weight. This is equivalent to the *average* aggregation of feature values covered by each bin, which shows to reduce misalignments in RoIAlign [18]:

$$\text{head}_i = \sum_{k=0}^{2\times 2} \sum_{j=0}^{\frac{m}{2}\times\frac{m}{2}} \frac{\alpha_k}{\frac{m}{2} \cdot \frac{m}{2}} v_{i_{k,j}}, \tag{11}$$

where a_k is the attention weight corresponding to k-th bin and $v_{i_{k,j}}$ is the j-th feature vector inside k-th bin.

Inspired by [8], we utilize the mask prediction of the previous decoder layer $\mathcal{M}_q \in \mathbb{R}^{H_m \times W_m}$ corresponding to the object query q. Given the coordinates of grid features within the region of interest r'_i, we sample the corresponding mask scores using bilinear interpolation. The sampled mask scores are binarized with the 0.5 threshold before $\mathrm{softmax}$ in the attention computation. Note that in masked instance-attention, we sample the feature grid of 14×14.

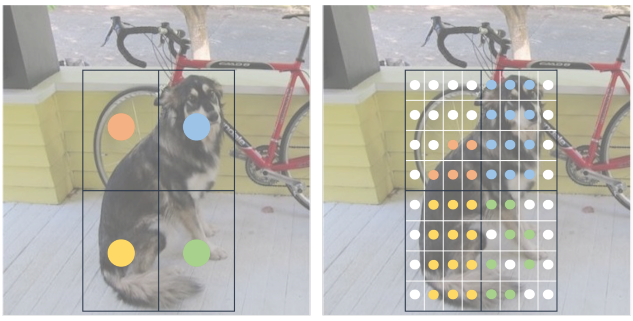


Figure 3. **Masked Instance-Attention. Left:** The box-attention [35] which samples 2×2 grid features in the region of interest. **Right:** Our masked instance-attention for dense grid sampling that employs masking strategy to capture object boundary. The 2×2 attention scores are denoted in four colours and the masked attention score is shown in white.

Fig. 3 shows the difference between box-attention [35] and masked instance-attention. By utilizing the mask prediction from previous decoder layer, masked instance-attention can effectively capture object of different shapes.

The creation of input features. In Fig. 4, we compare the creation of input features to detection head between SimpleFPN and our method. In [26], the multi-scale feature maps are created by different sets of convolution layers. Instead, SimPLR simply applies a deconvolution layer following by a GroupNorm layer [45].

Losses in training of SimPLR. We use focal loss [28] and dice loss [34] for the mask loss: $\mathcal{L}_{\text{mask}} = \lambda_{\text{focal}}\mathcal{L}_{\text{focal}} + \lambda_{\text{dice}}\mathcal{L}_{\text{dice}}$ with $\lambda_{\text{focal}} = \lambda_{\text{dice}} = 5.0$. The box loss is the combination of ℓ_1 loss and GIoU loss [38], $\mathcal{L}_{\text{box}} = \lambda_{\ell_1}\mathcal{L}_{\ell_1} + \lambda_{\text{giou}}\mathcal{L}_{\text{giou}}$, with $\lambda_{\ell_1} = 5.0$ and $\lambda_{\text{giou}} = 2.0$. The focal loss is also used for our classification loss, $\mathcal{L}_{\text{cls}}$. Our final loss is formulated as: $\mathcal{L} = \mathcal{L}_{\text{mask}} + \mathcal{L}_{\text{box}} + \lambda_{\text{cls}}\mathcal{L}_{\text{cls}}$ ($\lambda_{\text{cls}} = 2.0$ for object detection and instance segmentation, $\lambda_{\text{cls}} = 4.0$ for panoptic segmentation).

Hyper-parameters of SimPLR. SimPLR contains 6 encoder and decoder layers. The adaptive-scale attention in SimPLR encoder samples 2×2 grid features per region of interest. In the decoder, we compute attention on a grid of 14×14 features within regions of interest. The dimension ratio of feed-forward sub-layers to 4. The number of object queries is 300 in the decoder as suggested in [35]. The size of input image is 1024×1024 in both training and inference. Note that we also use this setting for the baseline (*i.e.*, BoxeR with ViT backbone).

In Tab. 2d, we show that the *decouple* between feature scale and dimension of the ViT backbone and the detection head helps to boost the performance of our plain detector while keeping the efficiency. This comes from the fact that the complexity of global self-attention in the ViT backbone increase quadraticaly w.r.t. the feature scale and the detection head enjoys the high-resolution input for object prediction. Note that with ViT-H as the backbone, we follow [26] to interpolate the kernel of patch projection into 16×16. The hyper-parameters for each SimPLR size (Base, Large, and Huge) are in Tab. 5.

model size		Base	Large	Huge
backbone	dim	768	1024	1280
	# head	12	16	16
	feat. scale	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$
detection head	enc. dim	384	768	960
	dec. dim	256	384	384
	# head	12	16	16
	feat. scale	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$

Table 5. Hyper-parameters of backbone and detection head for different sizes of SimPLR (base – large – huge models). Note that these settings are the same for all three tasks.

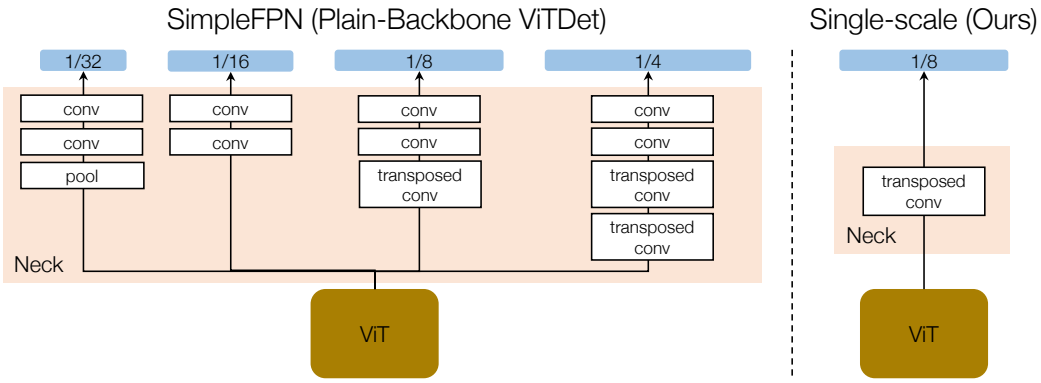


Figure 4. **The creation of input features. Left:** The creation of feature pyramids from the last feature of the plain backbone, ViT, in SimpleFPN [26] where different stacks of convolutional layers are used to create features at different scales. **Right:** The design of our single-scale feature map with only one layer.

B. Additional Results

method	backbone	pre-train	Panoptic Segmentation PQ	PQ^{th}	PQ^{st}	FPS
MaskFormer	Swin-B	sup-1K	51.1	56.3	43.2	-
Mask2Former	Swin-B	sup-1K	55.1	61.0	46.1	-
SimPLR	ViT-B	sup-1K	**55.2**	**61.2**	**46.2**	**13**

Table 6. **More panoptic segmentation comparison** between SimPLR with ViT-B backbone and other methods with Swin-B backbone. All backbones are pre-trained on ImageNet-1K with supervised pre-training. SimPLR still shows competitive results when using only single-scale input.

More panoptic segmentation comparison. Here, we provide more results of SimPLR with ViT-B backbone and other methods with Swin-B backbone using supervised pre-training on COCO panoptic segmentation in Tab. 6. SimPLR continues to show strong segmentation performance when using only single-scale input.

Ablation on pre-training strategies. Tab. 7 compares the ViT backbone when pre-trained using different strategies with different sizes of pre-training data. SimPLR with the ViT backbone benefits from better pre-training methods even with supervised approaches. Among supervised pre-training methods, DEiTv3 [41] shows better results than DEiT [40], and the pre-training on ImageNet-21K further improves the performance of DEiTv3.

However, self-supervised methods like MAE [19] provides strong pre-trained backbones when only pre-trained on ImageNet-1K. This further confirms that our plain detector, SimPLR, enjoys the significant progress of self-supervised learning and scaling ViTs. A similar observation is also pointed out in ViTDet [26] where the plain ViT backbone initialized with MAE shows better improvement over hierarchical backbones.

Figure 5. **Qualitative results** generated by SimPLR using ViT-B as backbone on the COCO `val` set. In each pair, the left image shows the visualization of instance detection and segmentation, while the right one indicates the panoptic segmentation.

pre-train	Object Detection				Instance Segmentation			
	AP^b	AP^b_S	AP^b_M	AP^b_L	AP^m	AP^m_S	AP^m_M	AP^m_L
IN-1K, DEiT	53.6	33.7	58.1	71.5	46.1	24.5	50.4	67.2
IN-1K, DEiTv3	54.0	34.3	58.8	70.5	46.4	24.8	51.1	66.7
IN-21K, DEiTv3	54.8	35.4	59.0	**72.4**	47.1	25.8	51.2	**68.5**
IN-1K, MAE	**55.4**	**36.1**	**59.1**	70.9	**47.6**	**26.8**	**51.4**	67.1

Table 7. **Ablation on pre-training strategies** of the plain ViT backbone using SimPLR evaluated on COCO object detection and instance segmentation. We compare the ViT backbone pre-trained using supervised methods (*top* row) *vs*. self-supervised methods (*bottom* row) with different sizes of pre-training dataset (ImageNet-1K *vs*. ImageNet-21K). Here, we use the 5× schedule as in [35]. It can be seen that SimPLR with the plain ViT backbone benefits from better pre-training approaches and with more pre-training data.

C. Qualitative results

We provide qualitative results of the SimPLR prediction with ViT-B backbone on three tasks: COCO object detection, instance segmentation, and panoptic segmentation in Fig. 5.

D. Asset Licenses

Dataset	License
ImageNet [10]	https://image-net.org/download.php
COCO [27]	Creative Commons Attribution 4.0 License

References

[1] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. In *ICLR*, 2022. 2

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Sandhini Agarwal Amanda Askell, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2020. 1, 2

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

[4] Mark Chen, Alec Radford, Rewon Child, Jeff Wu, Heewoo Jun, Prafulla Dhariwal, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, 2020. 2

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 2

[6] Wuyang Chen, Xianzhi Du, Fan Yang, Lucas Beyer, Xiaohua Zhai, Tsung-Yi Lin, Huizhong Chen, Jing Li, Xiaodan Song, Zhangyang Wang, and Denny Zhou. A simple single-scale vision transformer for object localization and instance segmentation. In *ECCV*, 2022. 2, 4, 8

[7] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *NeurIPS*, 2021. 2, 8

[8] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Mask2former: Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 1, 2, 4, 5, 6, 7, 8, 9

[9] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, and et al. Scaling vision transformers to 22 billion parameters. In *ICML*, 2023. 2, 5

[10] 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. 2, 11

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *ACL*, 2019. 1, 2

[12] Bin Dong, Fangao Zeng, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. SOLQ: Segmenting objects by learning queries. In *NeurIPS*, 2021. 2

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1, 2, 6

[14] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *ICCV*, 2021. 1, 7, 8

[15] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, TsungYi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *CVPR*, 2021. 6

[16] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 2, 9

[19] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 1, 2, 3, 5, 6, 7, 10

[20] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. In *ICCV*, 2021. 1

[21] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 2

[22] Ding Jia, Yuhui Yuan, Haodi He, Xiaopei Wu, Haojun Yu, Weihong Lin, Lei Sun, Chao Zhang, and Han Hu. Detrs with hybrid matching. In *CVPR*, 2023. 3

[23] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollar. Panoptic segmentation. In *CVPR*, 2019. 5

[24] Yann LeCun and Yoshua Bengio. *Convolutional Networks for Images, Speech and Time Series*, pages 255–258. MIT Press, 1995. 2

[25] Feng Li, Hao Zhang, Huaizhe xu, Shilong Liu, Lei Zhang, Lionel M. Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *CVPR*, 2023. 2

[26] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *ECCV*, 2022. 1, 2, 3, 4, 5, 6, 8, 9, 10

[27] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, 2014. 6, 11

[28] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 9

[29] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 1, 4

[30] Yutong Lin, Yuhui Yuan, Zheng Zhang, Chen Li, Nanning Zheng, and Han Hu. DETR does not need multi-scale or locality design. In *ICCV*, 2023. 1, 3, 6, 7, 8

[31] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. In *ECCV*, 2016. 4

[32] 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. In *ICCV*, 2021. 1, 2, 4, 7, 8

[33] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 6

[34] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3DV*, 2016. 9

[35] Duy-Kien Nguyen, Jihong Ju, Olaf Booij, Martin R. Oswald, and Cees G. M. Snoek. Boxer: Box-attention for 2d and 3d transformers. In *CVPR*, 2022. 1, 2, 3, 4, 5, 6, 7, 8, 9, 11

[36] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. BEiT v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv preprint arXiv:2208.06366*, 2022. 2, 5, 7

[37] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 2

[38] Seyed Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian D. Reid, and Silvio Savarese. Generalized intersection over union: A metric and A loss for bounding box regression. In *CVPR*, 2019. 9

[39] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2

[40] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers and distillation through attention. In *International Conference on Machine Learning*, 2021. 10

[41] Hugo Touvron, Matthieu Cord, and Herve Jegou. Deit iii: Revenge of the vit. *arXiv preprint arXiv:2204.07118*, 2022. 10

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1, 4

[43] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *CVPR*, 2021. 1, 2

[44] 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. In *ICCV*, 2021. 1

[45] Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018. 9

[46] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. 2

[47] Qihang Yu, Huiyu Wang, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. k-means mask transformer. In *ECCV*, 2022. 2

[48] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *CVPR*, 2022. 5

[49] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. In *ICLR*, 2023. 2

[50] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. K-net: Towards unified image segmentation. In *NeurIPS*, 2021. 1, 2

[51] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *ICLR*, 2021. 1, 2, 3, 4, 6