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

Anonymous Author(s)
Affiliation
Address
email

Abstract

This paper presents a *simple* and *plain* transformer-based architecture for detection and segmentation, we call SimPLR. Our network enables end-to-end prediction using the original Vision Transformer backbone without the need for complex modifications to accommodate multi-scale feature maps. By incorporating scale information into the attention computation, SimPLR effectively captures objects of different sizes. Remarkably, our study demonstrates that a single-scale feature map is adequate for our network, and with the proposed attention mechanism, SimPLR is capable of learning scale equivariant features. We show through our experiments that SimPLR, as a plain detector, achieves performance on par or better than its multi-scale or hierarchical counterparts. The straightforward end-to-end design of SimPLR is also universal as it yields competitive performance in object detection, instance segmentation, and panoptic segmentation while requiring only a small computational budget. The code will be released.

1 Introduction

After its astonishing success in natural language processing, the transformer [36] has quickly become the neural network architecture of choice in computer vision, as evidenced by recent success in image classification [28, 14], object detection [4, 42, 30] and segmentation [41, 8]. Different from natural language processing, where the same pre-trained network can be deployed for a wide range of downstream tasks with only minor modifications [2, 12], 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 object detection and instance 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.

The spatial nature of image data lies at the core of computer vision. Besides learning long range feature dependencies, the ability of capturing structure and relationships between neighboring pixels or features is critical for representing and understanding the image content. Building upon the successes of convolutional neural networks in computer vision, a line of research biases the transformer architecture to be *multi-scale* and *hierarchical* when dealing with the image input, (*i.e.*, Swin Transformer [28] and others [15, 37, 21]). They follow the design principles of convolutional neural networks by progressively increasing the receptive fields in each stage while spatially reducing the feature maps. This conventional design has proven effective in supporting multi-scale features for dense vision tasks and allows pre-trained transformers to be seamlessly integrated into convolution-based architectures with feature pyramid network [26], yielding impressive results in object detection and segmentation. Since the necessity of these complex adaptations in the transformer remains

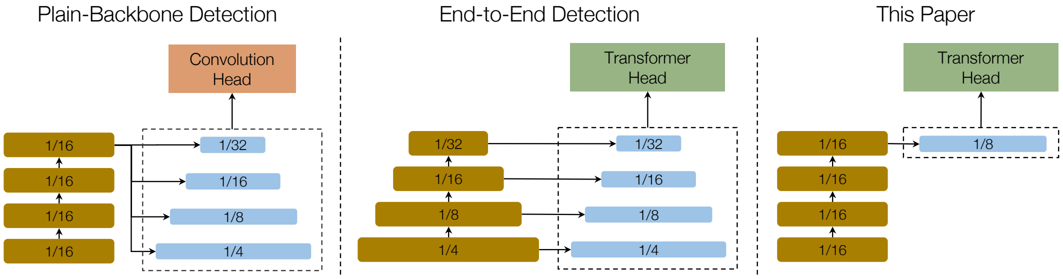

Figure 1: **Object detection architectures. Left:** The plain-backbone detector from [24] whose input (denoted in dashed region) are multi-scale features. **Middle:** state-of-the-art end-to-end detectors [30, 8] utilize a hierarchical backbone (*i.e.*, SwinTransformer [28]) to create multi-scale inputs. **Right:** Our simple single-scale detector following the end-to-end framework. Where existing detectors require multi-scale feature maps to be effective, we propose a plain detector whose backbone and detection head require only a single-scale feature.

uncertain, an alternative research direction pursues the idea of a simple transformer with "less inductive biases" and to emphasize learning vision-specific knowledge directly from image data. Starting with the end-to-end detection framework proposed in [4], a transformer-based prediction head removes many hand-designed components, like non-maximum suppression or intersection-over-union (IoU) computation, that encodes the prior knowledge. Recently, the Vision Transformer (ViT) [14], stands out as a *plain* and *non-hierarchical* architecture with a constant feature resolution, and acts as the feature extractor in plain-backbone detection. This is motivated by the success of ViTs in visual recognition [18, 1, 40, 10]. However, the plain design of ViTs casts doubts about its ability to capture information of objects across multiple scales. While recent studies [14, 24] suggest that ViTs with non-local self-attention computation could potentially learn translation-equivariant and scale-equivariant property during training, leading object detectors still require multi-scale feature maps. This observation holds true for both convolutional detectors [32, 19] and transformer-based detectors [42, 30, 8]. We postulate that an attention mechanism for scale-equivariant learning would enable the transformer-based detector to predict multi-scale objects using only the final feature map from a ViT backbone, thus eliminating the need for explicit multi-scale feature input.

With the goal of reducing inductive biases and emphasizing simplicity, we introduce SimPLR, an end-to-end object detector where both backbone and prediction head operate on plain features (See Fig. 1). Similar to [24], we focus on the *fine-tuning stage* and keep our approach independent of the pre-training phase. Our network architecture follows the recently introduced end-to-end object detection framework with transformer prediction head. In particular, SimPLR extracts the single-scale feature map from a pre-trained ViT which is then fed into the transformer encoder-decoder via a simple projection to make the final prediction. In order to capture objects of various sizes, we propose to incorporate scale information into the attention computation of the encoder, encouraging the network to learn scale-equivariant features. This eliminates the need for multi-scale feature maps and further simplifies the detection framework. For panoptic segmentation, our plain detector constructs a high-resolution feature map to capture fine-grained details and incorporates masked instance-attention in the decoder to improve object boundary delineation. We show the effectiveness of our architecture by experimental results on COCO object detection, instance segmentation, and panoptic segmentation. Notably, with a small computational budget, our plain detector shows competitive performance with leading detectors that employ hierarchical backbone and/or multi-scale feature map as input. We hope that our study of a simple and plain detector will open up more possibilities and flexibility for future research on dense vision and multi-modal vision models, where the alignment between visual regions and multi-modal tokens can be easily learnt via a plain feature map.

2 Method

Our goal is to further simplify the object detection pipeline from [24, 30], and to arrive at a universal object detection and segmentation model using plain features. To do so, we take advantage of recent progress in end-to-end object detection. Specifically, we adopt the box-attention mechanism [30] due to its effectiveness in learning discriminative object representations while being light-weight in

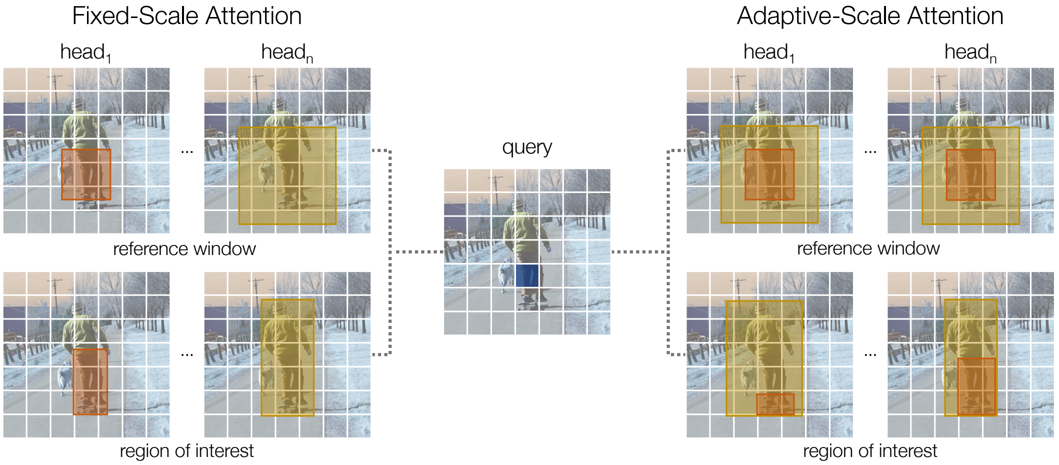

Figure 2: **Scale-aware attention mechanism. Left:** The fixed-scale attention with attention computation of a single scale per head. **Right:** The adaptive-scale attention with reference windows of multiple scales in each attention head. The 2-scale reference windows (denoted in yellow and orange) are used in this case. Unlike fixed-scale attention that transforms a single-scale reference window, adaptive-scale attention generates multi-scale regions of interest which are then selected using attention weights.

computation. We investigate the design of a plain detector[1] using *only a single-scale feature* from a plain transformer backbone during the *fine-tuning stage*.

In this section, we first review the box-attention mechanism. Then, the single-scale detector is presented for end-to-end object detection and segmentation. Finally, we propose several adaptations to make our model compatible with panoptic segmentation using a plain backbone.

2.1 Background

We first revisit box-attention proposed by Nguyen *et al.* [30]. Given the input feature map from the backbone, the encoder layers with box-attention will output contextual representations. The contextual representations are utilized to predict object proposals and to initialize object queries for the decoder. We denote the input feature map of an encoder layer as $e \in \mathbb{R}^{H_e \times W_e \times d}$ and the query vector $q \in \mathbb{R}^d$, with H_e, W_e, d denoting height, width, and dimension of the input features respectively. 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:

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

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

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

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.

Similar to self-attention [36], box-attention aggregates n multi-head features from regions of interest:

$$\text{MultiHeadAttention} = \text{Concat}(\text{head}_1, \dots, \text{head}_n)\, W^O \quad , \tag{4}$$

During the i-th attention head computation, a 2×2 feature grid is sampled from the corresponding region of interest r'_i, resulting in a set of value features $v_i \in \mathbb{R}^{2\times 2\times d_h}$. The 2×2 attention scores are efficiently generated by computing a dot-product between $q \in \mathbb{R}^d$ and relative position embeddings

[1] In this paper, "backbone" refers to the components that we inherit from the pre-training stage, "detection head" refers to the components that are initialized from scratch, and "plain" refers to the single-scale property. The "plain detector" is the detector whose backbone and detection head both operate on single-scale features.

($k_i \in \mathbb{R}^{2\times2\times d}$) followed by a $\mathrm{softmax}$ function. The attended feature $\mathrm{head}_i \in \mathbb{R}^{d_h}$ is a weighted average of the 2×2 value features in v_i with the corresponding attention weights:

$$\alpha = \mathrm{softmax}(q^\top k_i) \ , \tag{5}$$

$$\mathrm{head}_i = \mathrm{BoxAttention}(q, k_i, v_i) = \sum_{j=0}^{2\times2} \alpha_j v_{i_j} \ , \tag{6}$$

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 in [30] takes s multi-scale feature maps, $\{e^j\}_{j=1}^{s} (s = 4)$, as its inputs. In the i-th attention, s feature grids are sampled from each of multi-scale feature maps in order to produce head_i.

By computing the attended feature within regions of interest in each attention head, box-attention shows strong performance in end-to-end object detection with a small computational budget. The transformation functions (*i.e.*, translation and scaling) allow box-attention to capture long-range dependencies. The effectiveness and efficiency of box-attention leads us to a question whether we can learn multi-scale object information within a single-scale feature map.

2.2 SimPLR: A Simple and Plain Single-Scale Detector

Single-scale detector. Multi-scale feature maps have proven to be an effective solution to detect objects at multiple scales, from convolution-based object detectors [31, 32, 26] to transformer-based object detectors [42, 30, 24]. Multi-scale feature maps in a hierarchical backbone can be easily extracted from the pyramid structure [27, 26, 42]. 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 framework using ViTs remain unclear. Recent studies on plain-backbone detection [24, 7] conjecture the high-dimensional ViT with self-attention and positional embeddings [36] is able to preserve important information for localizing objects [2]. From this conjecture, we hypothesize that a properly designed transformer-based prediction head to encourage scale-equivariant learning is capable of capturing the necessary information for detecting multi-scale objects from a single-scale feature map.

We denote the *last* feature map from the plain backbone as $p \in \mathbb{R}^{H_p\times W_p\times d_p}$, where $H_p{=}\frac{H}{16}, W_p{=}\frac{W}{16}$ are the size of feature map, H, W are the size of the input image, and d_p is the hidden dimension of the backbone. The input feature, $e \in \mathbb{R}^{H_e\times W_e\times d}$, for the prediction head is simply created by a projection f, where H_e, W_e are the size of the input feature map and d is the hidden dimension of the prediction head. We consider the single-scale input feature map with different scales in $\{\frac{1}{32}, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}\}$. Different from [7], we empirically find that $H_e = \frac{H}{8}, W_e = \frac{W}{8}$ provides strong information for our detector to detect objects and helps to reduce the computational cost compared to higher resolution feature map. In the default setting ($H_e = \frac{H}{8}, W_e = \frac{W}{8}$), the projection f is a simple deconvolution layer.

In order to capture objects at different scales, 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 mechanism also leads to slow convergence in end-to-end detection [42]. Therefore, we prefer to adopt the box-attention in the encoder in order to learn multi-scale information in the input features.

Given the input feature map, $e \in \mathbb{R}^{H_e\times W_e\times d}$, and the query vector $q \in \mathbb{R}^d$, we assign multi-scale reference windows to q, denoted by $\hat{r}_t{=}[x, y, \hat{w}_t, \hat{h}_t]$, where the multi-scale reference windows corresponding to q are centered at the query spatial position. By default, we use 4-scale reference windows ($\hat{r}_1, ..., \hat{r}_4$) and set the size of t-th reference window to $\hat{w}_t = \hat{h}_t = 32 \cdot 2^{t-1}$ pixels. The use of multi-scale reference windows per query vector q is related to the multi-scale anchors in [32]. However, the multi-scale reference windows in our scenario are incorporated into multi-head attention mechanism in the encoder to learn scale-equivariant features, unlike [32], which utilizes them as the regression references in the final prediction layer. To incorporate scale information into multi-head attention, we study two scale-aware attention mechanisms (see Fig. 2):

[2] ViT-B and larger ($d \geq 768$) can maintain the information with a patch size of 16×16×3 in the input image.

i) *Fixed-Scale Attention.* We assign a reference window of one scale to each attention head. For example, with 8 attention heads and 4-scale reference windows, we assign each of attention heads with a reference window in a round robin manner, $r{=}\{\hat{r}_1, \hat{r}_2, \hat{r}_3, \hat{r}_4, \hat{r}_1, \hat{r}_2, \hat{r}_3, \hat{r}_4\}$. In the i-th attention head, we predict offset parameters w.r.t. to $r_i{=}[x, y, w_i, h_i]$:

$$\Delta_{x_i} = (qW_{x_i}^\top + b_{x_i}) * w_i \ \ , \qquad \Delta_{y_i} = (qW_{y_i}^\top + b_{y_i}) * h_i \ \ , \tag{7}$$

$$\Delta_{w_i} = \max(qW_{w_i}^\top + b_{w_i}, 0) * w_i \ \ , \qquad \Delta_{h_i} = \max(qW_{h_i}^\top + b_{h_i}, 0) * h_i \ \ , \tag{8}$$

where $W_{x_i}, W_{y_i}, W_{w_i}, W_{h_i}$ are learnable weights, $b_{x_i}, b_{y_i}, b_{w_i}, b_{h_i}$ are learnable biases in the i-th attention head. This is equivalent to a single attention operation per scale, which allows each attended feature, head_i, to capture a specific scale information w.r.t. the reference window r_i.

ii) *Adaptive-Scale Attention.* Here, each attention head needs to process reference windows of all scales ($\hat{r}_1, ..., \hat{r}_4$). In the i-th attention head, we predict 4 sets of offset parameters regarding to 4-scale reference windows. The corresponding 4 grid features, $v_i \in \mathbb{R}^{4\times2\times2\times d_h}$, are then sampled from the input feature map. The attention scores, $\alpha \in \mathbb{R}^{4\times2\times2}$, are generated for each feature in v_i in order to perform the attention operation. By computing the attention map on reference windows of multiple scales in each attention head, the attention mechanism is able to dynamically select necessary scale information for object prediction. We found that adaptive-scale attention leads to better performance.

Overall transformer architecture. Our network follows the design of a two-stage detector that performs end-to-end detection. Following [24], we simply use a plain ViT as the backbone with $14{\times}14$ windowed attention and four equally-distributed global attention blocks. Our prediction head consists of a transformer-based encoder and decoder for end-to-end detection as proposed in [4, 30]. We use output features from the encoder with scale-aware attention to produce object proposals using the last features of ViT backbone. The top-scored feature vectors will be initialized as object queries and the corresponding bounding box proposals will serve as its reference window in the decoder. The decoder utilizes instance-attention [30], an extension of box-attention for dense grid sampling, to decode object queries into bounding boxes for object detection and instance masks for instance segmentation. More details are provided in the supplementary document.

2.3 SimPLR for Universal Detection and Segmentation

To enable the plain detector on panoptic segmentation, we make an adaptation in the mask prediction which allows SimPLR to segment both "thing" and "stuff" categories effectively. To be specific, we predict segmentation masks of both types by computing the dot-product between object queries and a high-resolution feature map (*i.e.*, $\frac{1}{4}$ feature scale) following [8]. Our goal is to have *minimal* adaptation which allows us to use the same training recipe for all tasks, including object detection, instance segmentation, and panoptic segmentation. As the output feature of our encoder is in $\frac{1}{8}$ scale of the original image, we simply use a transposed convolution layer to generate the high-resolution feature for mask prediction. We found that it is beneficial to stack K encoder layers on top of the high-resolution feature to learn more contextual information. In addition, we incorporate the decoder with masked instance-attention to obtain discriminative representation for mask prediction. Different from masked cross-attention in [8], our masked instance-attention extends box-attention in [30] which yields a small complexity when dealing with feature map of $\frac{1}{8}$ scale from the encoder. More details are provided in the supplementary document.

Masked instance-attention. The masked instance-attention follows the grid sampling strategy of the box-attention in [30], 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 ($\mathbb{R}^d \rightarrow \mathbb{R}^{2\times2}$) 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 [19]:

$$\mathrm{head}_i = \sum_{k=0}^{2\times2} \sum_{j=0}^{\frac{m}{2}\times\frac{m}{2}} \frac{\alpha_k}{\frac{m}{2}\cdot\frac{m}{2}} \, v_{i_{k,j}} \, , \tag{9}$$

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.

plain-backbone detector	$\text{AP}^\text{b}\uparrow$	$\text{AP}^\text{m}\uparrow$	FLOPs↓
Multi-scale head			
ViTDet [24]	54.0	46.7	1.1T
BoxeR‡ [30]	55.4	47.7	0.5T
Single-scale head			
SimPLR	55.4	47.6	0.5T

Table 1: **Ablation on single-scale *vs*. multi-scale feature map** on COCO [25] object detection and instance segmentation using ViT-B as backbone. Compared to ViTDet, BoxeR performs better for both tasks with less FLOPs. Interestingly, SimPLR performs on par with BoxeR under the same FLOPs despite using only single-scale input (‡: BoxeR with SimpleFPN [24]).

Inspired by [8], we utilize the mask prediction before the sigmoid 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 and applied into the attention computation.

$$\text{head}_i = \sum_{k=0}^{2\times 2} \sum_{j=0}^{\frac{m}{2}\times\frac{m}{2}} \left(\frac{\alpha_k}{\frac{m}{2}\cdot\frac{m}{2}} + m_{i_{k,j}} \right) v_{i_{k,j}} \quad , \tag{10}$$

$$m_{i_{k,j}} = \begin{cases} 0 & \text{if sigmoid}\Big(\mathcal{M}_q\big(v_{i_{k,j}}\big)\Big) \geq 0.5 \\ -\infty & \text{otherwise} \end{cases} \quad , \tag{11}$$

where $\mathcal{M}_q\big(v_{i_{k,j}}\big)$ is the mask score sampled at the location of the feature $v_{i_{k,j}}$. The masked instance-attention along with high-resolution layers allow SimPLR to capture more fine-grained details, which is beneficial for panoptic segmentation.

3 Experiments

attention computation	$\text{AP}^\text{b}\uparrow$	$\text{AP}^\text{m}\uparrow$
base	54.5	46.8
i) fixed-scale attention	55.0	47.2
ii) adaptive-scale attention	55.4	47.6

(a) **Strategy for learning scale equivariance.**

feature scale	$\text{AP}^\text{b}\uparrow$	$\text{AP}^\text{m}\uparrow$
base	54.5	46.8
1/4	55.4	47.7
1/8	55.4	47.6
1/16	54.2	46.6

(b) **Scales of input feature map.**

window size	$\text{AP}^\text{b}\uparrow$	$\text{AP}^\text{m}\uparrow$
base	54.5	46.8
16	55.1	47.4
32	55.4	47.6
64	55.1	47.4

(c) **Reference window size.**

scale	$\text{AP}^\text{b}\uparrow$	$\text{AP}^\text{m}\uparrow$
base	54.5	46.8
2	54.6	47.0
4	55.4	47.6
6	55.2	47.6

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

Table 2: **Ablation on the design of attention mechanism** using a plain ViT backbone on COCO [25] object detection and instance segmentation. SimPLR receives the single-scale input from ViT and makes predictions. Compared to the naïve baseline which employs BoxeR and box-attention [30] with single-scale features, SimPLR improves performance of the plain detector.

Dataset, tasks & evaluation. In this study, we evaluate our method on COCO [25], a common detection and segmentation dataset, catering for object detection, instance segmentation, and panoptic segmentation tasks. The COCO dataset contains 118,000 training images and 5,000 validation images of 80 "thing" and 53 "stuff" categories. We report the average precision metric AP^b for object detection and AP^m for instance segmentation, and the panoptic quality metric PQ for panoptic segmentation. In panoptic segmentation, the union of "thing" and "stuff" categories are considered whereas in object detection and instance segmentation, the "thing" categories are the main focus. We train our network on the `train` split and evaluate on the `val` split.

Implementation details. By default, we use a single-scale feature map with adaptive-scale attention as described in Sec. 2.2. We initialize the backbone from MAE [18] pre-trained on ImageNet-1K without any labels. Unless otherwise specified, the hyper-parameters are the same as in [30]. For all experiments, our optimizer is AdamW [29] with a learning rate of 0.0001 and a weight decay

attention	PQ↑	PQ^{Th}↑	PQ^{St}↑
instance-attention [30]	54.9	61.2	45.3
masked instance-attention	55.3	61.6	45.8

(a) **Instance-attention *vs*. Masked instance-attention.**

K	PQ↑	PQ^{Th}↑	PQ^{St}↑
0	54.6	60.9	45.1
1	55.3	61.6	45.8

(b) **Number of high-resolution layers.**

Table 3: **Ablation on mask prediction adaptation** in a plain detector on COCO [25] panoptic segmentation. The masked instance-attention along with high-resolution layers improve the performance of SimPLR on panoptic segmentation, showing its effectiveness in capturing fine-grained information.

of 0.05. 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 [14] is set as the backbone. The layer-wise learning rate decay of 0.7 and drop path rate of 0.3 is applied to the ViT backbone. The input image size is 1024×1024 with large-scale jitter [16] 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 [16] with batch size of 16. Following [24], we also finetune our networks within 100 epochs and a batch size of 64 in the main experiments. As the main goal is to keep the architecture and training process simple, these settings are applied to all tasks (*i.e*., object detection, instance segmentation, and panoptic segmentation). The code will be released.

Ablation 1: A single-scale feature map is enough. In Tab. 1, we first compare plain-backbone detectors with single-scale *vs*. multi-scale features. The baseline of our study is the ViTDet: which employs the plain ViT as the backbone and Cascade R-CNN [3] with *multi-scale* feature maps from a simple feature pyramid (SimpleFPN) [24] as its detection head. We also train BoxeR with the plain backbone ViT and SimpleFPN using a $5\times$ schedule for a better comparison.

From our observation, the single-scale feature map is sufficient and our adaptive-scale attention mechanism enables SimPLR to perform detection using plain features. As a result, SimPLR matches the performance of BoxeR under the same amount of FLOPs. This observation is different from BoxeR and recent end-to-end detection approaches [42, 8] where a multi-scale feature map is needed to achieve competitive performance. We note that SimPLR follows a different perspective compared to convolution-based detection in [7] when using a single-scale feature map. The convolution-based detection operates region-of-interest (RoI) pooling on object proposals to encode multi-scale information. Instead, our transformer-based encoder performs attention computation per feature vector to learn scale equivariant features. The significance of multiple feature maps diminishes when the encoder is equipped with a powerful attention mechanism.

Ablation 2: Scale-aware attention mechanism is an important factor. Tab. 2 ablates the importance of the proposed attention mechanism. Here, we study the baseline which is the original box-attention from [30] with a single-scale feature map of $\frac{1}{8}$ scale and window size of 32 pixels (denoted in "base") and the design choices of our scale-aware attention mechanism for detection.

It can be seen from Tab. 2a that both attention strategies of encoding multi-scale information in attention computation demonstrate strong detection performance. By allowing the feature vector to choose suitable scale information in its attended feature, our adaptive-scale attention obtains an improvement of 0.9 AP point over the baseline.

In Tab. 2b, we verify the performance of SimPLR across multiple feature scales. The $\frac{1}{16}$ feature is created by applying a 1×1 convolution layer to the last ViT feature. For $\frac{1}{8}$ and $\frac{1}{4}$ feature, we use a transposed convolution layer. By default, we use SimPLR with adaptive-scale attention. Interestingly,

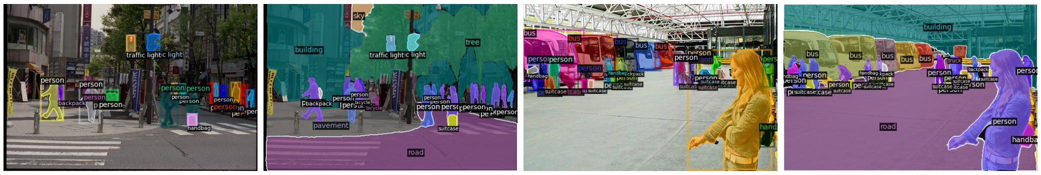

Figure 3: **Qualitative results** for object detection and panoptic segmentation on the COCO [25] 2017 `val` set generated by SimPLR. Note that SimPLR gives good predictions on small objects.

method	backbone	$\mathrm{AP^b}$	$\mathrm{AP^b_S}$	$\mathrm{AP^b_M}$	$\mathrm{AP^b_L}$	$\mathrm{AP^m}$	$\mathrm{AP^m_S}$	$\mathrm{AP^m_M}$	$\mathrm{AP^m_L}$	PQ	$\mathrm{PQ^{Th}}$	$\mathrm{PQ^{St}}$	FLOPs
Hierarchical, Multi-scale													
Swin† [28]	Swin-B	54.0	-	-	-	46.5	-	-	-	n/a			0.9T
MViT† [15]	MViT-B	55.6	-	-	-	**48.1**	-	-	-	n/a			0.8T
MaskFormer(5×) [9]	Swin-B	n/a				n/a				51.1	56.3	43.2	0.4T
Mask2Former(5×) [8]	Swin-B	n/a				46.7	26.1	50.5	**68.8**	55.1	61.0	**46.1**	0.5T
Plain, Multi-scale													
ViTDet [24]	ViT-B	54.0	-	-	-	46.7	-	-	-	n/a			1.1T
BoxeR‡(5×) [30]	ViT-B	55.4	-	-	-	47.7	-	-	-	n/a			0.5T
Simple, Plain													
UViT [7]	UViT-B	53.9	-	-	-	46.1	-	-	-	n/a			1.1T
SimPLR (5×)	ViT-B	55.4	36.1	59.1	70.9	47.6	26.8	51.4	67.1	55.3	61.6	45.8	0.5T
SimPLR	ViT-B	**55.6**	**37.1**	**59.2**	**71.6**	48.0	**27.5**	**51.5**	67.8	**55.6**	**62.1**	45.8	0.5T

Table 4: **Universal detection and segmentation comparison** for object detection, instance segmentation, and panoptic segmentation on the COCO [25] `val` set. By default, all backbones are pre-trained using ImageNet1K. (†: backbones pre-trained on ImageNet-22K; ‡: BoxeR with SimpleFPN from [24]; 5×: methods fine-tuned using 5× schedule; n/a: method is not applicable to the task). Methods in gray color are with a convolution-based detection head. SimPLR as a plain detector demonstrates competitive performance compared to many specialized or end-to-end models with a small amount of computation and is the only one suited for all three tasks.

the proposed method works well on $\frac{1}{16}$ features and performs worse than the baseline with $\frac{1}{8}$ features by a small gap of 0.3 AP point. The best trade-off between accuracy and efficiency comes from the features of $\frac{1}{8}$ scale. This results may align with DETR [4] where DETR-DC5 with dilated convolution in the C5 block shows better performance. In our case, we show that a simple transposed convolution layer is sufficient with plain ViT backbone.

Tab. 2c compares the performance of SimPLR across several base sizes of the reference window. Our attention mechanism has only marginal differences in term of the base size. The ablation reveals that the number of scales in attention computation rather than the reference window size plays an important role to make our network scale-aware. Indeed, in Tab. 2d, the use of multiple window scales shows improvement over the baseline. Increasing from 2 to 4 scales improves detection performance. This suggests that it is not trivial to be scale equivariant with a plain feature map and our adaptive-scale attention mechanism helps to strengthen features for detecting multi-scale objects.

Ablation 3: A single-scale architecture for panoptic segmentation. We evaluate the effectiveness of our adaptations on panoptic segmentation in Tab. 3. The masked instance-attention outperforms the instance attention, an extension of box-attention for dense grid in [30], when keeping the same efficiency. Moreover, adding encoder layer on top of high-resolution feature map delivers a large gain in panoptic segmentation. This evidence confirms the capability of the encoder layer in learning both fine-grained and semantically meaningful features.

Universal detection and segmentation. Tab. 4 lists the performance of SimPLR *vs*. previous methods in object detection, instance segmentation and panoptic segmentation. These approaches are divided into three parts: detectors with hierarchical backbone and multi-scale head, plain-backbone detectors using multi-scale features, and plain detector with both plain backbone and head. For plain-backbone methods, we report the performance of ViTDet [24] and BoxeR w/ SimpleFPN [30] fine-tuned on input image size of 1024 × 1024. Our method accounts for less FLOPs than convolution-based detection head. Across different detectors, SimPLR achieves strong performance on COCO object detection and instance segmentation. Interestingly, SimPLR outperforms Mask2Former [8] in detecting small and medium objects but lags behind in large category. This makes SimPLR to be a competitive plain detector. In panoptic segmentation, we observe the same behaviour in which SimPLR improves over "thing" categories but lags behind in "stuff" categories. Overall, SimPLR, as a plain detector, can achieve highly accurate results on all three tasks and can compete with hierarchical and multi-scale detectors. Visualizations of SimPLR predictions are shown in Fig. 3.

Limitations. Our final goal is to simplify the detection pipeline and to achieve competitive results at the same time. In Tab. 1 and Tab. 2, we find that the attention mechanism which can effectively learn scale-aware information in its computation plays a key role for a plain detector. However, our adaptive-scale attention mechanism still encodes the knowledge of different scales. In the future, we hope that with the help of 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.

4 Related Work

Backbones for object detection. Inspired by R-CNN [17], modern object detection methods utilize a task-specific head on top of a pre-trained backbone for their prediction. Initially, object detectors were dominated by convolutional neural network (CNN) backbones [23] pre-trained on ImageNet [11], with consistent accuracy and efficiency improvements over subsequent generations of architectures, *e.g.* [34, 38, 20, 22]. By removing the need of labels, methods with self-supervised learning have emerged as an even more powerful solution for pre-training general vision representations [6, 18]. With the success of the transformer in learning from large-scale text data [2, 12], many studies have explored the transformer for computer vision [5, 14]. Recently, the Vision Transformer (ViT) [14] demonstrated the capability of learning meaningful representation for visual recognition. We show through experiments that the pre-trained ViT when combined with a transformer-based prediction head is able to learn scale-equivariant features throughout the architecture.

Multi-scale object detection. A key challenge in object detection is to detect objects within an image across multiple scales. Early works tackled this problem by applying a CNN detector with a sliding window strategy on an image pyramid [33] or with a generated region proposal [35] to extract each scale-normalized region from the input image [17]. To reduce computational costs, Faster R-CNN [32] generates object proposals on the feature map using a region proposal network. As deeper features of CNNs tends to capture more high-level information at the expense of fine-grained details for small objects, SSD [27] predicts objects from multiple layers of the feature hierarchy. The Feature Pyramid Network [26] further improves the creation of multi-scale feature maps with top-down fusion and lateral connections. With the recent evidence that non-hierarchical transformer architectures (*i.e.*, ViTs) are able to learn convolution-like behaviour through image data (*i.e.*, translation-equivariance), the necessity of multi-scale feature maps becomes questionable. Recent studies on plain-backbone detection [24, 7] show that the last feature of ViTs captures the necessary information to detect objects at multiple scales. However, these detectors still rely on multi-scale features to achieve better performance [24]. We remove this requirement in the prediction head, allowing us to rely on a simple and plain transformer for both the backbone and the task-specific head.

End-to-end detection and segmentation. A universal architecture solves multiple vision tasks without many architectural changes. This is made possible by the introduction of transformers for object detection [4]. Follow-up works [13, 30] extended the end-to-end detection framework to object detection and instance segmentation. This inspired MaskFormer [9] and K-Net [41] to unify segmentation tasks with a class-agnostic mask prediction. Pointing out that MaskFormer and K-Net still lag behind specialized architectures, Cheng *et al.* introduce Mask2Former [8], which outperforms specialized architectures on instance, semantic, and panoptic segmentation. Most recently, Yu *et al.* [39] replace self-attention with k-means clustering, further boosting the effectiveness of the network. Similar to convolution-based detection heads, these approaches utilize multi-scale feature maps from a hierarchical backbone. In this work, we remove the multi-scale feature map constraint and enable a universal architecture using a *single-scale* feature map from a plain vision transformer.

5 Conclusion

We presented SimPLR, a simple and plain detector that eliminates the requirement for 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. By utilizing only single-scale features as input, SimPLR achieves highly competitive performance on *multiple* tasks on COCO: object detection, instance segmentation, and panoptic segmentation. These findings suggest that the plain detector holds great promise, opening up opportunities for further simplifying the architecture design. The use of a plain feature map throughout SimPLR shows the potential for multi-task or multi-modal learning. We hope this study could encourage the further exploration of simple neural network architectures for dense vision tasks.

References

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

[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.

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: High quality object detection and instance segmentation. *TPAMI*, 2019.

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

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

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

[7] 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.

[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.

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

[10] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd van Steenkiste, Gamaleldin F. Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Alexey Gritsenko Mark Patrick Collier, Vighnesh Birodkar, Cristina Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetić, Dustin Tran, Thomas Kipf, Mario Lučić, Xiaohua Zhai, Daniel Keysers, Jeremiah Harmsen, and Neil Houlsby. Scaling vision transformers to 22 billion parameters. *arXiv preprint arXiv:2302.05442*, 2023.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

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

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

[14] 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.

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

[16] 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.

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

[18] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

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

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

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

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

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

[24] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *ECCV*, 2022.

[25] 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.

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

[27] 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.

[28] 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.

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

[30] 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.

[31] Pedro H. O. Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollár. Learning to refine object segments. In *ECCV*, 2016.

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

[33] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *ICLR*, 2014.

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

[35] Jasper R.R. Uijlings, Koen E.A. van de Sande, Theo Gevers, and Arnold W.M. Smeulders. Selective search for object recognition. *IJCV*, 2013.

[36] 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.

[37] 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.

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

[39] 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.

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

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

[42] 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.