Dynamic Head: Unifying Object Detection Heads with Attentions

Xiyang Dai   Yinpeng Chen   Bin Xiao   Dongdong Chen   Mengchen Liu
Lu Yuan   Lei Zhang
Microsoft
Redmond, USA

{xidai, yiche, bixi, dochen, mengcliu, luyuan, leizhang}@microsoft.com

Abstract

The complex nature of combining localization and classification in object detection has resulted in the flourished development of methods. Previous works tried to improve the performance in various object detection heads but failed to present a unified view. In this paper, we present a novel dynamic head framework to unify object detection heads with attentions. By coherently combining multiple self-attention mechanisms between feature levels for scale-awareness, among spatial locations for spatial-awareness, and within output channels for task-awareness, the proposed approach significantly improves the representation ability of object detection heads without any computational overhead. Further experiments demonstrate that the effectiveness and efficiency of the proposed dynamic head on the COCO benchmark. With a standard ResNeXt-101-DCN backbone, we largely improve the performance over popular object detectors and achieve a new state-of-the-art at 54.0 AP. Furthermore, with latest transformer backbone and extra data, we can push current best COCO result to a new record at 60.6 AP. The code will be released at https://github.com/microsoft/DynamicHead.

1. Introduction

Object detection is to answer the question "what objects are located at where" in computer vision applications. In the deep learning era, nearly all modern object detectors [11, 23, 12, 35, 28, 31, 33] share the same paradigm – a backbone for feature extraction and a head for localization and classification tasks. How to improve the performance of an object detection head has become a critical problem in existing object detection works.

The challenges in developing a good object detection head can be summarized into three categories. Firstly, the head should be *scale-aware*, since multiple objects with vastly distinct scales often co-exist in an image. Secondly, the head should be *spatial-aware*, since objects usually ap-

pear in vastly different shapes, rotations, and locations under different viewpoints. Thirdly, the head needs to be *task-aware*, since objects can have various representations (*e.g*., bounding box [12], center [28], and corner points [33]) that own totally different objectives and constraints. We find recent studies [12, 35, 28, 31, 33] only focus on solving one of the aforementioned problems in various ways. It remains an open problem how to develop a unified head that can address all these problems simultaneously.

In this paper, we propose a novel detection head, called *dynamic head*, to unify scale-awareness, spatial-awareness, and task-awareness all together. If we consider the output of a backbone (*i.e*., the input to a detection head) as a 3-dimensional tensor with dimensions $level \times space \times channel$, we discover that such a unified head can be regarded as an attention learning problem. An intuitive solution is to build a full self-attention mechanism over this tensor. However, the optimization problem would be too difficult to solve and the computational cost is not affordable.

Instead, we can deploy attention mechanisms separately on each particular dimension of features, *i.e*., level-wise, spatial-wise, and channel-wise. The scale-aware attention module is only deployed on the dimension of $level$. It learns the relative importance of various semantic levels to enhance the feature at a proper level for an individual object based on its scale. The spatial-aware attention module is deployed on the dimension of $space$ (*i.e*., $height \times width$). It learns coherently discriminative representations in spatial locations. The task-aware attention module is deployed on $channels$. It directs different feature channels to favor different tasks separately (*e.g*., classification, box regression, and center/key-point learning.) based on different convolutional kernel responses from objects.

In this way, we explicitly implement a unified attention mechanism for the detection head. Although these attention mechanisms are separately applied on different dimensions of a feature tensor, their performance can complement each other. Extensive experiments on the MS-COCO benchmark

demonstrate the effectiveness of our approach. It offers a great potential for learning a better representation that can be utilized to improve all kinds of object detection models with $1.2\% \sim 3.2\%$ AP gains. With the standard ResNeXt-101-DCN backbone, the proposed method achieves a new state of the art 54.0% AP on COCO. Besides, compared with EffcientDet [27] and SpineNet [8], dynamic head uses $1/20$ training time, yet with a better performance. Furthermore, with latest transformer backbone and extra data from self-training, we can push current best COCO result to a new record at 60.6 AP (see appendix for details).

2. Related Work

Recent studies focus on improving object detectors from various perspectives: scale-awareness, spatial-awareness and task-awareness.

Scale-awareness. Many researches have empathized the importance of scale-awareness in object detection as objects with vastly different scales often co-exist in natural images. Early works have demonstrated the significance of leveraging image pyramid methods [6, 24, 25] for multi-scale training. Instead of image pyramid, feature pyramid [15] was proposed to improve efficiency by concatenating a pyramid of down-sampled convolution features and had become a standard component in modern object detectors. However, features from different levels are usually extracted from different depth of a network, which causes a noticeable semantics gap. To solve this discrepancy, [18] proposed to enhance the features in lower layers by bottom-up path augmentation from feature pyramid. Later, [20] improved it by introducing balanced sampling and balanced feature pyramid. Recently, [31] proposed a pyramid convolution to extract scale and spatial features simultaneously based on a modified 3-D convolution.

In this work, we present a scale-aware attention in the detection head, which makes the importance of various feature level adaptive to the input.

Spatial-awareness. Previous works have tried to improve the spatial-awareness in object detection for better semantic learning. Convolution neural networks were known to be limited in learning spatial transformations existed in images [41]. Some works mitigate this problem by either increasing the model capability (size) [13, 32] or involving expensive data augmentations [14], resulting in extremely high computational cost in inference and training. Later, new convolution operators were proposed to improve the learning of spatial transformations. [34] proposed to use dilated convolutions to aggregate contextual information from the exponentially expanded receptive field. [7] proposed a deformable convolution to sample spatial locations with ad-

ditional self-learned offsets. [37] reformulated the offset by introducing a learned feature amplitude and further improved its ability.

In this work, we present a spatial-aware attention in the detection head, which not only applies attention to each spatial location, but also adaptively aggregates multiple feature levels together for learning a more discriminative representation.

Task-awareness. Object detection was originated from a two-stage paradigm [39, 6], which first generates object proposals and then classifies the proposals into different classes and background. [23] formalized the modern two-stage framework by introducing Region Proposal Networks (RPN) to formulate both stages into a single convolution network. Later, one-stage object detector [22] became popular due to its high efficiency. [16] further improved the architecture by introducing task-specific branches to surpass the accuracy of two-stage detectors while maintaining the speed of previous one-stage detectors.

Recently, more works have discovered that various representations of objects could potentially improve the performance. [12] first demonstrated that combining bounding box and segmentation mask of objects can further improve the performance. [28] proposed to use center representations to solve object detection in a per-pixel prediction fashion. [35] further improved the performance of center-based method by automatically selecting positive and negative samples according to statistical characteristics of object. Later, [33] formulated object detection as representative key-points to ease the learning. [9] further improved the performance by detecting each object as a triplet, rather than a pair of key-points to reduce incorrect predictions. Most recently, [21] proposed to extract border features from the extreme points of each border to enhance the point feature and archived the state-of-the-art performance.

In this work, we present a task-aware attention in the detection head, which allows attention to be deployed on channels, which can adaptively favor various tasks, for either single-/two-stage detectors, or box-/center-/keypoint-based detectors.

More importantly, all the above properties are integrated into a unified attention mechanism in our head design. To our best knowledge, it is the first general detection head framework which takes a step towards understanding what role attention plays in the success of object detection head.

3. Our Approach

3.1. Motivation

In order to enable scale-awareness, spatial-awareness and task-awareness simultaneously in a unified object de-
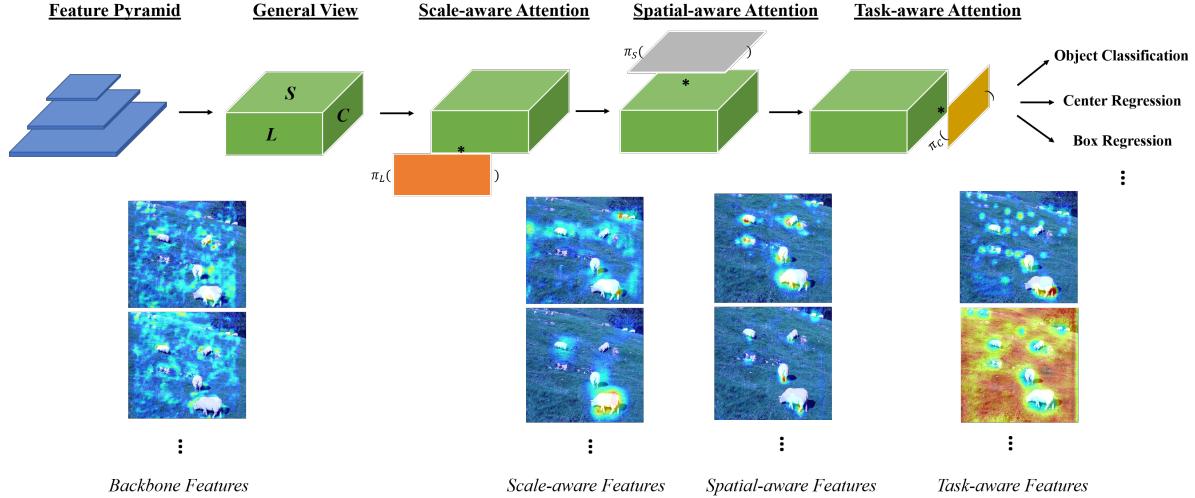
Figure 1. An illustration of our Dynamic Head approach. It contains three different attention mechanisms, each focusing on a different perspective: scale-aware attention, spatial-aware attention, and task-aware attention. We also visualize how the feature maps are improved after each attention module.

tection head, we need to generally understand previous improvements on object detection heads.

Given a concatenation of features $\mathcal{F}_{in} = \{F_i\}_{i=1}^{L}$ from L different levels in a feature pyramid, we can resize consecutive level features towards the scale of the median level feature using up-sampling or down-sampling. The re-scaled feature pyramid can be denoted as a 4-dimensional tensor $\mathcal{F} \in \mathcal{R}^{L \times H \times W \times C}$, where L represents the number of levels in the pyramid, H, W, and C represent height, width, and the number of channels of the median level feature respectively. We further define $S = H \times W$ to reshape the tensor into a 3-dimensional tensor $\mathcal{F} \in \mathcal{R}^{L \times S \times C}$. Based on this representation, we will explore the role of each tensor dimension.

- The discrepancy of object scales is related to features at various levels. Improving the representation learning across different levels of $\mathcal{F}$ can benefit scale-awareness of object detection.
- Various geometric transformations from dissimilar object shapes are related to features at various spatial locations. Improving the representation learning across different spatial locations of $\mathcal{F}$ can benefit spatial-awareness of object detection.
- Divergent object representations and tasks can be related to the features at various channels. Improving the representation learning across different channels of $\mathcal{F}$ can benefit task-awareness of object detection.

In this paper, we discover that all above directions can be unified in an efficient attention learning problem. Our work is the first attempt to combine multiple attentions on all three dimensions to formulate a unified head for maximizing their improvements.

3.2. Dynamic Head: Unifying with Attentions

Given the feature tensor $\mathcal{F} \in \mathcal{R}^{L \times S \times C}$, the general formulation of applying self-attention is:

$$W(\mathcal{F}) = \pi(\mathcal{F}) \cdot \mathcal{F} \qquad (1)$$

where $\pi(\cdot)$ is an attention function. A naïve solution to this attention function is implemented by fully connected layers. But directly learning the attention function over all dimensions is computationally costly and practically not affordable due to the high dimensions of the tensor.

Instead, we convert the attention function into three sequential attentions, each focusing on only one perspective:

$$W(\mathcal{F}) = \pi_C \left(\pi_S \left(\pi_L(\mathcal{F}) \cdot \mathcal{F} \right) \cdot \mathcal{F} \right) \cdot \mathcal{F}, \qquad (2)$$

where $\pi_L(\cdot)$, $\pi_S(\cdot)$, and $\pi_C(\cdot)$ are three different attention functions applying on dimension L, S, and C, respectively.

Scale-aware Attention π_L. We first introduce a scale-aware attention to dynamically fuse features of different scales based on their semantic importance.

$$\pi_L(\mathcal{F}) \cdot \mathcal{F} = \sigma \left(f \left(\frac{1}{SC} \sum_{S,C} \mathcal{F} \right) \right) \cdot \mathcal{F} \qquad (3)$$

where $f(\cdot)$ is a linear function approximated by a 1×1 convolutional layer, and $\sigma(x) = max(0, min(1, \frac{x+1}{2}))$ is a hard-sigmoid function.

Spatial-aware Attention π_S. We apply another spatial-aware attention module based on the fused feature to focus on discriminative regions consistently co-existing among both spatial locations and feature levels. Considering the high dimensionality in S, we decompose this module into two steps: first making the attention learning sparse by using deformable convolution [7] and then aggregating features across levels at the same spatial locations:

$$\pi_S(\mathcal{F})\cdot\mathcal{F} = \frac{1}{L}\sum_{l=1}^{L}\sum_{k=1}^{K}w_{l,k}\cdot\mathcal{F}(l;p_k+\Delta p_k;c)\cdot\Delta m_k, \quad (4)$$

where K is the number of sparse sampling locations, $p_k + \Delta p_k$ is a shifted location by the self-learned spatial offset Δp_k to focus on a discriminative region and Δm_k is a self-learned importance scalar at location p_k. Both are learned from the input feature from the median level of $\mathcal{F}$.

Task-aware Attention π_C. To enable joint learning and generalize different representations of objects, we deploy a task-aware attention at the end. It dynamically switches ON and OFF channels of features to favor different tasks:

$$\pi_C(\mathcal{F})\cdot\mathcal{F} = max\bigg(\alpha^1(\mathcal{F})\cdot\mathcal{F}_c+\beta^1(\mathcal{F}), \alpha^2(\mathcal{F})\cdot\mathcal{F}_c+\beta^2(\mathcal{F})\bigg), \quad (5)$$

where $\mathcal{F}_c$ is the feature slice at the c-th channel and $[\alpha^1, \alpha^2, \beta^1, \beta^2]^T = \theta(\cdot)$ is a hyper function that learns to control the activation thresholds. $\theta(\cdot)$ is implemented similar to [3], which first conducts a global average pooling on $L \times S$ dimensions to reduce the dimensionality, then uses two fully connected layers and a normalization layer, and finally applies a shifted sigmoid function to normalize the output to $[-1, 1]$.

Finally, since the above three attention mechanisms are applied sequentially, we can nest Equation 2 multiple times to effectively stack multiple π_L, π_S, and π_C blocks together. The detailed configuration of our dynamic head (*i.e.*, *Dy-Head* for simplification) block is shown in Figure 2 (a).

As a summary, the whole paradigm of object detection with our proposed dynamic head is illustrated in Figure 1. Any kinds of backbone network can be used to extract feature pyramid, which is further resized to the same scale, forming a 3-dimensional tensor $\mathcal{F} \in \mathcal{R}^{L\times S\times C}$, and then used as the input to the dynamic head. Next, several Dy-Head blocks including scale-aware, spatial-aware, and task-aware attentions are stacked sequentially. The output of the dynamic head can be used for different tasks and representations of object detection, such as classification, center/box regression, etc..

At the bottom of Figure 1, we show the output of each type of attention. As we can see, the initial feature maps from backbones are noisy due to the domain difference from
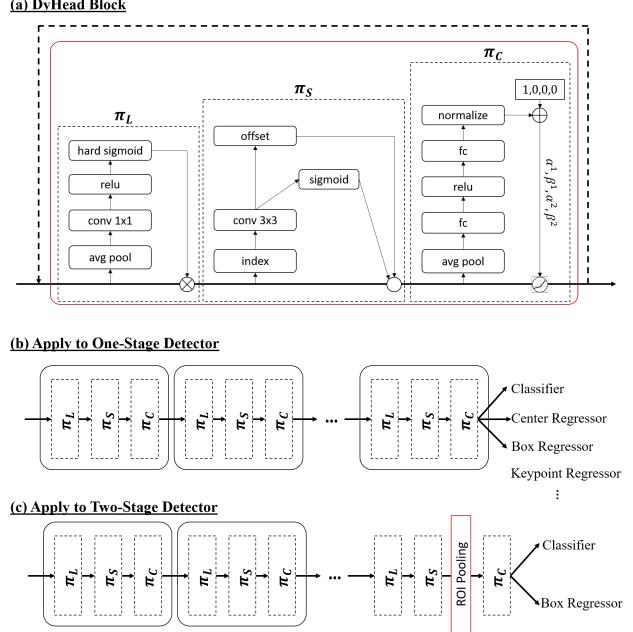

Figure 2. A detailed design of Dynamic Head. (a) shows the detailed implementation of each attention module. (b) shows how to apply our dynamic head blocks to one-stage object detector. (c) shows how to apply our dynamic head blocks to two-stage object detector.

ImageNet pre-training. After passing through our scale-aware attention module, the feature maps become more sensitive to the scale differences of foreground objects; After further passing through our spatial-aware attention module, the feature maps become more sparse and focused on discriminative spatial locations of foreground objects. Finally, after passing through our task-aware attention module, the feature maps re-form into different activations based on the requirements of different down-stream tasks. These visualizations well demonstrate the effectiveness of each attention module.

3.3. Generalizing to Existing Detectors

In this section, we demonstrate how the proposed dynamic head can be integrated into existing detectors to effectively improve their performances.

One-stage Detector. One-stage detector predicts object locations by densely sampling locations from feature map, which simplifies the detector design. Typical one-stage detector (*e.g.*, RetinaNet [16]) is composed of a backbone network to extract dense features and multiple task-specific sub-network branches to handle different tasks separately. As shown in previous work [3], object classification sub-network behaves very differently from bounding box re-

gression sub-network. Controversial to this conventional approach, we only attach one unified branch instead of multiple branches to the backbone. It can handle multiple tasks simultaneously, thanks to the advantage of our multiple attention mechanisms. In this way, the architecture can be further simplified and the efficiency is improved as well. Recently, anchor-free variants of one-stage detectors became popular, for example, FCOS [28], ATSS [35] and RepPoint [33] re-formulated objects as centers and key-points to improve performance. Compared to RetinaNet, these methods require to attach a centerness prediction, or a keypoint prediction to either the classification branch or the regression branch, which makes the constructions of task-specific branches non-trivial. By contrast, deploying our dynamic head is more flexible since it only appends various types of predictions to the end of head, shown in Figure 2 (b).

Two-stage Detector. Two-stage detectors utilize region proposal and ROI-pooling [23] layers to extract intermediate representations from feature pyramid of a backbone network. To cooperate this characteristic, we first apply our scale-aware attention and spatial-aware attention on feature pyramid before a ROI-pooling layer and then use our task-aware attention to replace the original fully connected layers, as shown in Figure 2 (c).

3.4. Relation to Other Attention Mechanisms

Deformable. Deformable convolution [7, 37] has significantly improved the transformation learning of traditional convolutional layers by introducing sparse sampling. It has been widely used in object detection backbones to enhance the feature representations. Although it is rarely utilized in object detection head, we can regard it as solely modeling the S sub-dimension in our representation. We find the deformable module used in the backbone can be complementary to the proposed dynamic head. In fact, with the deformable variant of ResNext-101-64x4d backbone, our dynamic head achieves a new state-of-the-art object detection result.

Non-local. Non-Local Networks [30] is a pioneer work of utilizing attention modules to enhance the performance of object detection. However, it uses a simple formulation of dot-product to enhance a pixel feature by fusing other pixels' features from different spatial locations. This behavior can be regarded as modeling only the $L \times S$ sub-dimensions in our representation.

Transformer. Recently, there is a trend to introduce the Transformer module [29] from natural language processing into computer vision tasks. Preliminary works [2, 38, 5] have demonstrated promising results in improving object

detection. Transformer provides a simple solution to learn cross-attention correspondence and fuse features from different modalities by applying multi-head fully connected layers. This behavior can be viewed as modeling only the $S \times C$ sub-dimensions in our representation.

The aforementioned three types of attention works only partially model sub-dimensions in the feature tensor. As a unified design, our dynamic head combines attentions on different dimensions into one coherent and efficient implementation. The following experiments show such a dedicated design can help existing object detectors achieve remarkable gains. Besides, our attention mechanisms explicitly address the challenges of object detection, in contrast to implicit working principles in existing solutions.

4. Experiment

We evaluate our approach on the MS-COCO dataset [17] following the commonly used settings. MS-COCO contains 80 categories of around 160K images collected from the web. The dataset is split into the train2017, val2017, and test2017 subsets with 118K, 5K, 41K images respectively. The standard mean average precision (AP) metric is used to report results under different IoU thresholds and object scales. In all our experiments, we only train on the train2017 images without using any extra data. For experiments of ablation studies, we evaluate the performances on the val2017 subset. When comparing to state-of-the-art methods, we report the official result returned from the test server on test-dev subset.

4.1. Implementation Details

We implement our dynamic head block as a plugin, based on the popular implementation of Mask R-CNN benchmark [12]. If it is not specifically mentioned, our dynamic head is trained with the ATSS framework [35] . All models are trained using one compute node of 8 V100 GPUs each with 32GB memory.

Training. We use ResNet-50 as the model backbone in all ablation studies and train it with the standard 1x configuration. Other models are trained with the standard 2x training configurations as introduced in [12]. We use an initial learning rate of 0.02 with weight decay of $1e-4$ and momentum of 0.9 . The learning rate is stepped down by a factor of 0.1 at the 67% and 89% of training epochs. Standard augmentation with random horizontal flipping is used. To compare with previous methods trained with multi-scale inputs, we also conduct multi-scale training for selective models.

Inference. To compare with state-of-the-art methods reported using test time augmentation, we also evaluate our

best model with multi-scale testing. Other tricks, such as model EMA, mosaic, mix-up, label smoothing, soft-NMS or adaptive multi-scale testing [25], are not used.

4.2. Ablation Study

We conduct a series of ablation studies to demonstrate the effectiveness and efficiency of our dynamic head.

L.	S.	C.	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
×	×	×	39.0	57.2	42.4	22.1	43.1	50.2
✓	×	×	39.9	57.8	43.5	25.4	44.0	52.4
×	✓	×	41.4	58.5	45.2	26.8	45.2	54.3
×	×	✓	40.3	58.3	43.9	24.2	44.6	53.7
×	✓	✓	42.0	59.5	45.5	25.5	46.1	55.2
✓	×	✓	40.6	58.6	44.4	24.6	44.8	53.3
✓	✓	×	41.9	59.2	45.6	24.8	46.1	54.5
✓	✓	✓	**42.6**	**60.1**	**46.4**	**26.1**	**46.8**	**56.0**

Table 1. Ablation study on the effectiveness of each attention module in our dynamic head block.

Effectiveness of Attention Modules. We first conduct a controlled study on the effectiveness of different components in our dynamic head block by gradually adding them to the baseline. As shown in Table 1, "L.", "S.", "C." represent our scale-aware attention module, spatial-aware attention module, and task-aware module, respectively. We can observe that individually adding each component to the baseline implementation improves its performance by 0.9 AP, 2.4 AP and 1.3 AP. It is expected to see the spatial-aware attention module archives the biggest gain because of its dominant dimensionality among three modules. When we add both "L." and "S" to the baseline, it continuously improves the performance by 2.9 AP. Finally, our full dynamic head block significantly improves the baseline by 3.6 AP. This experiment demonstrates that different components work as a coherent module.

Effectiveness on Attention Learning. We then demonstrate the effectiveness of attention learning in our dynamic head module. Figure 3 shows the trend of the learned scale ratios (calculated by dividing the learned weight of higher resolution by the learned weight of lower resolution) on different level of features in our scale-aware attention module. The histogram is calculated using all images from the COCO val2017 subset. It is clear to see that our scale-aware attention module tends to regulate higher resolution feature maps ("level 5" purple histogram in the figure) toward lower resolution and lower resolution feature maps ("level 1" blue histogram in the figure) toward higher resolution to smooth
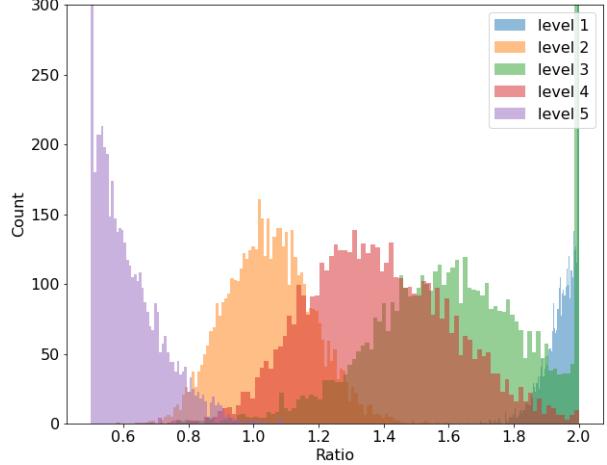

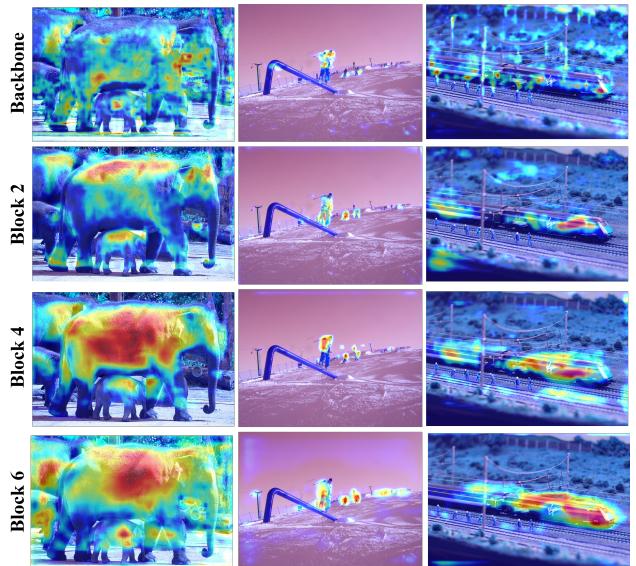Figure 3. Ablation study on the effectiveness of our scale-aware attention module.

Figure 4. A visualization on the effectiveness of our spatial-aware attention module.

the scale discrepancy form different feature levels. This proves the effectiveness of scale-aware attention learning.

Figure 4 visualizes the feature map output before and after applying different number (i.e. 2,4,6) of blocks of attention modules. Before applying our attention modules, the feature maps extracted from the backbone are very noisy and fail to focus on the foreground objects. As the feature maps pass through more attention modules (from block 2 to block 6 as shown in the figure), it is obvious to see the feature maps cover more foreground objects and focus more accurately on their discriminative spatial locations. This visualization well demonstrates the effectiveness of the spatial-aware attention learning

#Block	GFLOPs	AP	AP_{50}	AP_{75}
Baseline	254.39	39.0	57.2	42.4
1	-84.69	36.7	55.5	40.0
2	-63.45	39.5	57.8	43.1
4	-20.97	42.0	59.9	45.9
6	+21.50	**42.6**	**60.1**	**46.4**
8	+63.98	42.5	59.6	46.1
10	+106.46	42.3	59.4	45.9

Table 2. Ablation study on the efficiency and effectiveness of stacking different number of dynamic head blocks.

Efficiency on the Depth of Head. We evaluate the efficiency of our dynamic head by controlling the depth (number of blocks). As shown in Table 2, we vary the number of used DyHead blocks (*e.g.*, 1, 2, 4, 8, 10 blocks) and compare their performances and computational costs (GFLOPs) with the baseline. Our dynamic head can benefit from the increase of depth by stacking more blocks until 8. It is worth noting that our method with 2 blocks has already outperformed the baseline at even lower computation cost. Meanwhile, even with 6 blocks, the increment of computational cost is negligible compared to the computation cost of the backbone, while largely improving the accuracy. It demonstrates the efficiency of our method.

Generalization on Existing Object Detectors. We evaluate the generalization ability of the dynamic head by plugging it to popular object detectors, such as Faster-RCNN [23], RetinaNet [16], ATSS [35], FCOS [28], and RepPoints [33]. These methods represent a wide variety of object detection frameworks (*e.g.*, two-stage vs. one-stage, anchor-based vs. anchor-free, box-based vs. point-based). As shown in Table 3, our dynamic head significantly boosts all popular object detectors by $1.2 \sim 3.2\ AP$. It demonstrates the generality of our method.

4.3. Comparison with the State of the Art

We compare the performance of the dynamic head with several standard backbones and state-of-the-art object detectors.

Cooperate with Different Backbones. We first demonstrate the compatibility of dynamic head with different backbones. As shown in Table 4, we evaluate the performances of object detector by integrating dynamic head with the ResNet-50, ResNet-101 and ResNeXt-101 backbones, and compare with recent methods with similar configurations, including Mask R-CNN [12], Cascade-RCNN [1], FCOS [28], ATSS [35] and BorderDet [21]. Our method consistently outperforms previous methods with a big margin. When compared to the best detector BorderDet [21]

Method	AP	AP_{50}	AP_{75}
anchor-based two-stage:			
Faster R-CNN [23]	36.4	57.9	39.4
+ DyHead	**38.9**	**57.6**	**42.0**
anchor-based one-stage:			
RetinaNet [16]	35.7	54.3	37.9
+ DyHead	**38.4**	**57.5**	**41.3**
anchor-free box-based:			
ATSS [35]	39.4	57.5	42.9
+ DyHead	**42.6**	**60.1**	**46.4**
anchor-free center-based:			
FCOS [28]	38.8	57.3	41.9
+ DyHead	**40.0**	**58.2**	**43.4**
anchor-free keypoint-based:			
RepPoints [33]	38.2	59.7	40.7
+ DyHead	**39.6**	**59.8**	**42.8**

Table 3. Ablation study on the generalization of our dynamic head when applying to popular object detection methods.

with same settings, our method outperforms it by $1.1\ AP$ with the ResNet-101 backbone and by $1.2\ AP$ with the ResNeXt-64x4d-101 backbone, where the improvement is significant due to the challenges in the COCO benchmark.

Compared to State-of-the-Art Detectors. We compare our methods with state-of-the-art detectors [35, 31, 21, 4, 2, 27, 8], including some concurrent works [38, 5]. As shown in Table 5, we summarize these existing work into two categories: one using multi-scale training, and the other using both multi-scale training and multi-scale testing.

Compared with methods with only multi-scale training, our method achieves a new state of the art at $52.3\ AP$ with only 2x training schedule. Our method is competitive and more efficient to learn compared with EffcientDet [27] and SpineNet [8], with a significantly less $1/20$ training time. Compared with the latest work [2, 38, 5], which utilize Transformer modules as attention, our dynamic head is superior to these methods with more than $2.0\ AP$ gain, while using less training time than theirs. It demonstrates that our dynamic head can coherently combine multiple modalities of attentions from different perspectives into a unified head, resulting in better efficiency and effectiveness.

We further compare our method with state-of-the-art results [35, 21, 4, 38, 5] with test time augmentation (TTA) using both multi-scale training and multi-scale testing. Our dynamic head helps achieve a new state-of-the-art result at $54.0\ AP$, which significantly outperforms concurrent best methods by $1.3\ AP$.

Method	Backbone	Iteration	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
two-stage detector:								
Mask R-CNN[12]	ResNet-101	2x	38.2	60.3	41.7	20.1	41.1	50.2
Cascade-RCNN[1]	ResNet-50	3x	40.6	59.9	44.0	22.6	42.7	52.1
Cascade-RCNN[1]	ResNet-101	3x	42.8	62.1	46.3	23.7	45.5	55.2
one-stage detector:								
FCOS[28]	ResNet-101	2x	41.5	60.7	45.0	24.4	44.8	51.6
FCOS[28]	ResNeXt-64x4d-101	2x	43.2	62.8	46.6	26.5	46.2	53.3
ATSS[35]	ResNet-101	2x	43.6	62.1	47.4	26.1	47.0	53.6
ATSS[35]	ResNeXt-64x4d-101	2x	45.6	64.6	49.7	28.5	48.9	55.6
BorderDet[21]	ResNet-101	1x	43.2	62.1	46.7	24.4	46.3	54.9
BorderDet[21]	ResNet-101	2x	45.4	64.1	48.8	26.7	48.3	56.5
BorderDet[21]	ResNeXt-64x4d-101	2x	46.5	65.7	50.5	29.1	49.4	57.5
DyHead	ResNet-50	1x	43.0	60.7	46.8	24.7	46.4	53.9
DyHead	ResNet-101	2x	46.5	64.5	50.7	28.3	50.3	57.5
DyHead	ResNeXt-64x4d-101	2x	47.7	65.7	51.9	31.5	51.7	60.7

Table 4. Comparison with results using different backbones on the MS COCO test-dev set

Method	Backbone	Iteration	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
multi-scale training:								
ATSS[35]	ResNeXt-64x4d-101-DCN	2x	47.7	66.5	51.9	29.7	50.8	59.4
SEPC[31]	ResNeXt-64x4d-101-DCN	2x	50.1	69.8	54.3	31.3	53.3	63.7
BorderDet[21]	ResNeXt-64x4d-101-DCN	2x	48.0	67.1	52.1	29.4	50.7	60.5
RepPoints v2[4]	ResNeXt-64x4d-101-DCN	2x	49.4	68.9	53.4	30.3	52.1	62.3
RelationNet++[5]	ResNeXt-64x4d-101-DCN	2x	50.3	69.0	55.0	32.8	55.0	65.8
DETR[2]	ResNet-101	~25x	44.9	64.7	47.7	23.7	49.5	62.3
Deformable DETR[38]	ResNeXt-64x4d-101-DCN	~4x	50.1	69.7	54.6	30.6	52.8	64.7
EfficientDet[27]	Efficient-B7	~50x	52.2	71.4	56.3	–	–	–
SpineNet[8]	SpineNet-190	~40x	52.1	71.8	56.5	35.4	55.0	63.6
DyHead	ResNeXt-64x4d-101-DCN	2x	**52.3**	**70.7**	**57.2**	**35.1**	**56.2**	**63.4**
multi-scale training and multi-scale testing:								
ATSS[35]	ResNeXt-64x4d-101-DCN	2x	50.7	68.9	56.3	33.2	52.9	62.4
BorderDet[21]	ResNeXt-64x4d-101-DCN	2x	50.3	68.9	55.2	32.8	52.8	62.3
RepPoints v2[4]	ResNeXt-64x4d-101-DCN	2x	52.1	70.1	57.5	34.5	54.6	63.6
Deformable DETR[38]	ResNeXt-64x4d-101-DCN	~4x	52.3	71.9	58.1	34.4	54.4	65.6
RelationNet++[5]	ResNeXt-64x4d-101-DCN	2x	52.7	70.4	58.3	35.8	55.3	64.7
DyHead	ResNeXt-64x4d-101-DCN	2x	**54.0**	**72.1**	**59.3**	**37.1**	**57.2**	**66.3**

Table 5. Comparison with the state-of-the-art results on the MS COCO test-dev set

5. Conclusion

In this paper, we have presented a novel object detection head, which unify the scale-aware, spatial-aware, and task-aware attentions in a single framework. It suggests a new view of object detection head with attentions. As a plugin block, the dynamic head can be flexibly integrated into any existing object detector framework to boost its performance. Moreover, it is efficient to learn. Our study shows that designing and learning attentions in the object detection head is an interesting direction which deserves more focused studies. This work only takes a step, and could be further improved in these aspects: how to make the full attention model easy to learn and efficient to compute, and how to systematically consider more modalities of attentions into the head designing for better performance.

Method	Backbone	Iteration	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$
Mask R-CNN[12]	Swin-T	3x	46.0	68.1	50.3	31.2	49.2	60.1
Cascade Mask R-CNN[1]	Swin-T	3x	50.4	69.2	54.7	33.8	54.1	65.2
RepPoints v2[4]	Swin-T	3x	50.0	68.5	54.2	–	–	–
SparseRCNN[26]	Swin-T	3x	47.9	67.3	52.3	–	–	–
ATSS[35]	Swin-T	3x	47.2	66.5	51.3	–	–	–
DyHead	Swin-T	2x	49.7	68.0	54.3	33.3	54.2	64.2

Table 6. Comparison with results using transformer backbone on the MS COCO validation set.

Method	Backbone	Iteration	AP$_{val}$	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$
CenterNet2† [36]	Res2Net-101-DCN	8x	56.1	56.4	74.0	61.6	38.7	59.7	68.6
CopyPaste† [10]	Efficient-B7	8x	57.0	57.3	–	–	–	–	–
HTC++[19]	Swin-L	6x	58.0	58.7	–	–	–	–	–
DyHead	Swin-L	2x	58.4	58.7	77.1	64.5	41.7	62.0	72.8
DyHead†	Swin-L	2x	**60.3**	**60.6**	**78.5**	**66.6**	**43.9**	**64.0**	**74.2**

Table 7. Comparison with latest methods on the MS COCO test-dev set. † demonstrates method with extra data.

Appendix

We keep improving our performance after submission. Recently, there is a hot trend on adapting transformers as vision backbones and demonstrating promising performance. When training our dynamic head with latest transformer backbone [19], extra data and increased input size, we can further improve the current SOTA on COCO benchmark.

Cooperate with Transformer Backbones. We cooperate our dynamic head with the latest transformer-based backbones, such as [19]. Shown in Table 6, our dynamic head is competitive to [1] which requires extra mask ground-truth to help boost performance. Meanwhile, compared to the baseline method [35] used in our framework, we further improve its performance by $2.5\ AP$. This well proves that our dynamic head is complementary to transformer-based backbone to further improve its performance on downstream object detection task.

Cooperate with Larger Inputs and Extra Data. We find that our dynamic head can further benefit from larger input size and extra data generated using self-training method [40]. We increase the maximum image side from 1333 to 2000 and use a multi-scale training with minimum image side varying from 480 to 1200. Similar to the training scheme described in section 4.1, we avoid using more tricks to ensure reproducibility. As shown in Table 7, our dynamic head leads significant gain compared to latest works [10, 36] and matches the performance of [19] without using extra mask ground-truth. Meanwhile, our dynamic head requires less than $1/3$ of training time of these works. This demonstrates our superior efficiency and effectiveness. Fur-

thermore, we follow [40] to generate pseudo labels on ImageNet dataest and use it as an extra data. Our dynamic head can largely benefit from large scale data and further improve the COCO state-of-the-art result to a new record high at $60.6\ AP$.

References

[1] Zhaowei Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6154–6162, 2018. 7, 8, 9

[2] Nicolas Carion, F. Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *ArXiv*, abs/2005.12872, 2020. 5, 7, 8

[3] Y. Chen, X. Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic relu. *ArXiv*, abs/2003.10027, 2020. 4

[4] Y. Chen, Zheng Zhang, Yue Cao, L. Wang, Stephen Lin, and H. Hu. Reppoints v2: Verification meets regression for object detection. *ArXiv*, abs/2007.08508, 2020. 7, 8, 9

[5] Cheng Chi, Fangyun Wei, and Han Hu. Relationnet++: Bridging visual representations for object detection via transformer decoder. *ArXiv*, abs/2010.15831, 2020. 5, 7, 8

[6] Jifeng Dai, Y. Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. *ArXiv*, abs/1605.06409, 2016. 2

[7] Jifeng Dai, Haozhi Qi, Y. Xiong, Y. Li, Guodong Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 764–773, 2017. 2, 4, 5

[8] Xianzhi Du, Tsung-Yi Lin, Pengchong Jin, Golnaz Ghiasi, Mingxing Tan, Yin Cui, Quoc V. Le, and Xiaodan Song. Spinenet: Learning scale-permuted backbone for recognition and localization, 2020. 2, 7, 8

[9] Kaiwen Duan, S. Bai, Lingxi Xie, H. Qi, Q. Huang, and Q. Tian. Centernet: Keypoint triplets for object detection. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6568–6577, 2019. 2

[10] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D. Cubuk, Quoc V. Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation, 2020. 9

[11] Ross B. Girshick. Fast r-cnn. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015. 1

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. 1, 2, 5, 7, 8, 9

[13] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 2

[14] A. Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *CACM*, 2017. 2

[15] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017. 2

[16] Tsung-Yi Lin, Priyal Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:318–327, 2020. 2, 4, 7

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2014. cite arxiv:1405.0312Comment: 1) updated annotation pipeline description and figures; 2) added new section describing datasets splits; 3) updated author list. 5

[18] Shu Liu, Lu Qi, Haifang Qin, J. Shi, and J. Jia. Path aggregation network for instance segmentation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8759–8768, 2018. 2

[19] 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, 2021. 9

[20] Jiangmiao Pang, K. Chen, J. Shi, H. Feng, Wanli Ouyang, and D. Lin. Libra r-cnn: Towards balanced learning for object detection. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 821–830, 2019. 2

[21] Han Qiu, Yuchen Ma, Zeming Li, Songtao Liu, and J. Sun. Borderdet: Border feature for dense object detection. In *ECCV*, 2020. 2, 7, 8

[22] Joseph Redmon, S. Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016. 2

[23] Shaoqing Ren, Kaiming He, Ross B. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015. 1, 2, 5, 7

[24] B. Singh and L. Davis. An analysis of scale invariance in object detection - snip. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3578–3587, 2018. 2

[25] B. Singh, Mahyar Najibi, and L. Davis. Sniper: Efficient multi-scale training. In *NeurIPS*, 2018. 2, 6

[26] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, and Ping Luo. SparseR-CNN: End-to-end object detection with learnable proposals. *arXiv preprint arXiv:2011.12450*, 2020. 9

[27] Mingxing Tan, R. Pang, and Quoc V. Le. Efficientdet: Scalable and efficient object detection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10778–10787, 2020. 2, 7, 8

[28] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9626–9635, 2019. 1, 2, 5, 7, 8

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, L. Kaiser, and Illia Polosukhin. Attention is all you need. *ArXiv*, abs/1706.03762, 2017. 5

[30] X. Wang, Ross B. Girshick, A. Gupta, and Kaiming He. Non-local neural networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018. 5

[31] Xinjiang Wang, S. Zhang, Zhuoran Yu, Litong Feng, and Wayne Zhang. Scale-equalizing pyramid convolution for object detection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13356–13365, 2020. 1, 2, 7, 8

[32] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5987–5995, 2017. 2

[33] Ze Yang, S. Liu, H. Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9656–9665, 2019. 1, 2, 5, 7

[34] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *CoRR*, abs/1511.07122, 2016. 2

[35] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Z. Lei, and S. Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9756–9765, 2020. 1, 2, 5, 7, 8, 9

[36] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Probabilistic two-stage detection. In *arXiv preprint arXiv:2103.07461*, 2021. 9

[37] X. Zhu, H. Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results.

2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 9300–9308, 2019. 2, 5

[38] X. Zhu, Weijie Su, Lewei Lu, Bin Li, X. Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *ArXiv*, abs/2010.04159, 2020. 5, 7, 8

[39] C. Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014. 2

[40] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. Rethinking pretraining and self-training. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 9

[41] Zhengxia Zou, Z. Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *ArXiv*, abs/1905.05055, 2019. 2