Dynamic DETR: End-to-End Object Detection with Dynamic Attention

Xiyang Dai Yinpeng Chen Jianwei Yang Pengchuan Zhang
Lu Yuan Lei Zhang
Microsoft
{xidai, yiche, jianwyan, penzhan, luyuan, leizhang}@microsoft.com

Abstract

In this paper, we present a novel Dynamic DETR (Detection with Transformers) approach by introducing dynamic attentions into both the encoder and decoder stages of DETR to break its two limitations on small feature resolution and slow training convergence. To address the first limitation, which is due to the quadratic computational complexity of the self-attention module in Transformer encoders, we propose a dynamic encoder *to approximate the Transformer encoder's attention mechanism using a convolution-based dynamic encoder with various attention types. Such an encoder can dynamically adjust attentions based on multiple factors such as scale importance, spatial importance, and representation (*i.e.*, feature dimension) importance. To mitigate the second limitation of learning difficulty, we introduce a* dynamic decoder *by replacing the cross-attention module with a ROI-based dynamic attention in the Transformer decoder. Such a decoder effectively assists Transformers to focus on region of interests from a coarse-to-fine manner and dramatically lowers the learning difficulty, leading to a much faster convergence with fewer training epochs. We conduct a series of experiments to demonstrate our advantages. Our Dynamic DETR significantly reduces the training epochs (by* **14**×*), yet results in a much better performance (by* **3.6** *on mAP). Meanwhile, in the standard* 1× *setup with ResNet-50 backbone, we archive a new state-of-the-art performance that further proves the learning effectiveness of the proposed approach.*

1. Introduction

Object detection aims at predicting a set of bounding boxes and category labels for each object of interest. Modern object detectors are based on convolutional neural networks, and share the same paradigm – a backbone for feature extraction and a head for localization and classification tasks [22, 10]. Until recently, Detection Transformer (DETR) has been proposed as an alternative solution to the object detection problem. It views object detection as a set-based matching problem. By leveraging Transformers [25] originally developed for language tasks, it is able to model the relations of objects and their global image context from a set of learned object queries. It then performs a global optimization that forces unique predictions from object queries via bipartite matching, effectively removing the need of hand-designed components such as non-maximum suppression (NMS) and anchor generation in traditional object detection methods.

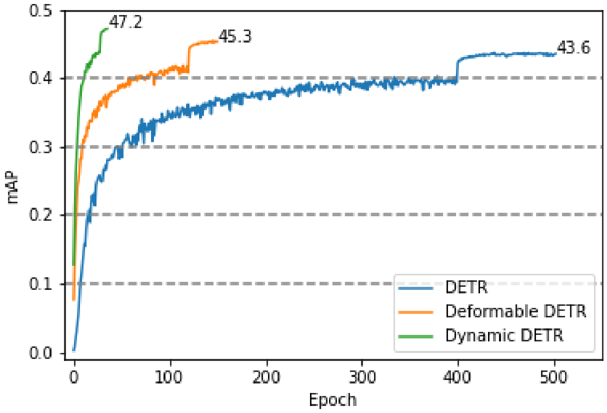

Figure 1. Convergence curve comparison between our proposed approach and state-of-the-art methods. Our Dynamic DETR largely reduces the training epochs (by 14×), yet results in a significantly better performance (by 3.6).

However, DETR suffers from several problems that prevent it from wide adoption in the community. On one hand, the input resolution of features maps is limited in the native Transformer as feature encoder, since the complexity of the self-attention module grows quadratically with the increase of the input resolution. It results in incompatibility to the typical feature pyramid that is widely used in modern object detectors, and relatively low performance at detecting small objects. On the other hand, it requires much longer training epochs to converge than the existing object detectors since the cross-attention module struggles to learn on a large global feature map from an initial dense attention to a final sparse attention. Thus, it is the high demand of an effective solution for improving DETR.

Recent work Deformable DETR [29], which combines the sparse spatial sampling of deformable convolution, and the relation modeling capability of Transformers, to mitigate the slow convergence and high complexity issues of DETR. It has achieved noticeable improvements on performance and efficiency in training. It is interesting to exploit if the efficiency and performance of DETR can be further improved.

In this paper, we propose an alternative solution to address the above two problems of DETR by a dynamic attention framework, called *Dynamic DETR*, which consists of a dynamic encoder and a dynamic decoder. We replace the Transformer encoder in DETR with a new convolution-based dynamic encoder, which apply dynamic attention on full scales of feature pyramid based on scale importance, spatial importance, and representation (*i.e*., feature dimension) importance. Since it makes self-attentions feasible on full scale of representations from low to high resolutions, the performance of DETR can be significantly boosted. In addition, the dynamic decoder replaces the cross-attention module in the DETR decoder with a ROI-based dynamic attention, which can effectively assists Transformers to focus on regions of interest in a coarse-to-fine manner and dramatically lowers the learning difficulty, leading to a much faster convergence with fewer training epochs.

Our contribution can be summarized in three-folds:

- We propose a novel *Dynamic DETR* approach, which coherently combines a dynamic convolution-based encoder and a dynamic Transformer-based decoder. The proposed approach significantly improves the representation ability of object detection head and the learning efficiency without any computational overhead.

- Compared to the original DETR, Our Dynamic DETR largely reduces the training epochs (by 14×), yet results in a significantly better performance (by 3.6), shown in Figure 1.

- To our best knowledge, we are the first end-to-end method that achieves a better than traditional performance in the standard 1x setup with a ResNet-50 backbone, at 42.9 mAP.

2. Background

Feature Pyramid. Recognizing objects at vastly different scales that co-existed in natural images is a fundamental challenge in computer vision. Researches have explored many directions to incorporate multiple scale into object detection. [6] first introduced image pyramid to object detection by independently computing features on different scales and regions of the images separately. Later, [22, 21] discarded such idea and only unitized single scale features due to the slow speed of extracting features from multiple scales of images. To overcome the obvious disadvantages of image pyramid, feature pyramid [14] was proposed, which combined both the feature hierarchy computed from a neural network and a top down architecture with lateral connections to enhance the semantics at all levels. It largely improved the performance without sacrificing efficiency and had hence become a standard component in modern object detectors. However, features from different levels were usually extracted from different depths of the network, which causes a semantic gap. Recent works continuously to improve the feature pyramid by introducing more comprehensive architectures. [17] introduced a bottom-up path augmentation from the feature pyramid to enhance the features in lower layers. Later, [19] further improved it by introducing balanced feature pyramid, together with balanced sampling and and balancing loss to mitigate the adverse effects caused by the imbalance in feature level. Recently, [26] proposed to extract scale and spatial features simultaneously in the spirit of 3D convolution to aggregate contextual information at different levels. Most recently, [8] proposed to unify scale-awareness, spatial-awareness, and task-awareness together by applying multiple attention mechanisms on feature pyramid and resulted in a significant improvement.

Dynamic Attention. Recently, researchers began to incorporate dynamic attention mechanisms to CNN to enhance its feature representation. [12] first proposed a novel "Squeeze-and-Excitation" (SE) unit, which adaptively recalibrates channel-wise feature responses by explicitly modeling inter-dependencies between channels. [7] proposed a deformable convolution to sample spatial locations with additional self-learned offsets. [28] reformulated the offset by introducing a learned feature amplitude and further improved its ability. [3] proposed to aggregate multiple parallel convolution kernels dynamically based upon their attentions, which had more representation power since these kernels are aggregated in a nonlinear way via attention. [4] introduced a dynamic ReLU of which parameters were generated by a hyper function over input elements. It encodes the global context into the hyper function and adapts the piecewise linear activation function accordingly based on the input. Most recently, there is a trend to adapt Transformer [25] from natural language processing into vision tasks. Its multi-head self-attention and cross-attention mechanisms provide a powerful way to model long-range and cross-modal dependencies, which are highly desired properties in a wide range of computer vision tasks, such as image classification, object detection, and so on.

End-to-End Object Detection. The basic formulation of object detection has been settled into either one-stage or two-stage methods, which have not been changed for a while. [6, 22] formalized the modern two-stage object detection by first introducing Region Proposal Networks (RPN) to extract region features and then applying a second stage to refine the prediction. Meanwhile, [21, 15] introduced the one-stage object detector by directly regressing bounding boxes and predicting class probabilities from convolutional features of full image in a single neural network, which results in high efficiency. Furthermore, [1] introduced cascade procedure to form a multiple-stage detector. [23] later improved the performance by learning a sparse set of object proposals instead of dense anchor priors.

Alternatively, Detection Transformer (DETR) [2] was proposed to provide an alternative solution to object detection. It presents an end-to-end optimization objective for set prediction and formulates the loss function via a bipartite matching mechanism between object proposals and ground-truth labels. It adapts an Transformer encoder-decoder head built upon the CNN backbone. Such an approach effectively removes the need of hand-designed components such as non-maximum suppression (NMS) and anchor generation in traditional object detection methods. However, it did not solve a few problems when adapting from language tasks to vision tasks such as limited input feature resolutions and much longer training epochs. Most recently, Deformable DETR[29] was proposed to solve such problems by introducing deformbale convolutions into the Transformer framework and achieved significant improvements.

This work presents an alternative solution to address the existing problems of DETR and further boost the performance and training efficiency achieved by Deformable DETR to a new high level. By contrast, our key contribution consists of a convolution-based dynamic encoder and an improved Transformer-based dynamic decoder. Our approach is able to apply dynamic attention on full scales of feature pyramid varying from low to high resolutions and assist Transformers to focus on regions of interest in a coarse-to-fine manner to enable faster convergence.

3. Dynamic DETR

In this section, we study the existing problems in DETR at first. Then we introduce dynamic encoder and dynamic decoder in our Dynamic DETR respectively to address these problems.

3.1. Revisiting DETR

The most crucial attention module in Detection Transformer (DETR) is the multi-head attention layer directly from Transformer [25] used in language processing. Given a query Q, a key K and a value V, the multi-head attention layer models attentions from different representation subspaces and positions in parallel using multiple heads $\mathcal{H}$:

$$\texttt{MultiHeadAttn}(Q, K, V) = \texttt{Concat}(\mathcal{H}_i, \ldots, \mathcal{H}_m)W^O$$
$$\mathcal{H}_i = \texttt{softmax}\left(\frac{QW_i^Q(KW_i^K)^T}{\sqrt{d_k}}VW_i^V\right) \tag{1}$$

where d_k is the dimension of the key, m is the number of heads, W_i^Q, W_i^K, W_i^V are linear projections and W^O is the projection matrix to fuse features from different heads together. Such an attention layer formulates the self-attention and cross-attention mechanisms by varying the combination of key and value.

When applied to object detection task, a few compromises have been made to adapt the spatial information. In the Transformer encoder stage, only the self-attention mechanism is preset. The query and key are flattened pixels from feature maps, which makes the complexity to be quadratic with respect to the input spatial scales. This largely limits the resolution of input feature maps and makes extracting features from pyramid representation infeasible. In the Transformer decoder stage, the multi-head attention layer has been formulated in both the self-attention and cross-attention mechanisms. The burden of self-attention has been released as the key and query are from learn-able object embeddings. However, in the cross-attention mechanism, the key is still fed from feature maps, which causes the learning difficulty of attending a query to a sparse locally-focused region from an initial uniform attention on the whole feature maps.

To overcome these above problems, we propose to apply dynamic attentions in both the encoder stage to take advantages of feature pyramid representations and the decoder stage to accelerate the training convergence, as shown in Figure 2.

3.2. Dynamic Encoder

In contrast to directly improving Transformer encoders, we seek to use a convolution-based approach to approximate the self-attention mechanisms. Given a set of features $P = \{P_1, \ldots, P_k\}$ ($k = 5$ for typical object detectors) from a feature pyramid, ideally, we would like to have a multi-scale self-attention function π:

$$\texttt{MultiScaleSelfAttn}(P) = \pi(P) \cdot P \tag{2}$$

Unfortunately, this is infeasible due to the varied scales of feature maps from the feature pyramid. Inspired by the

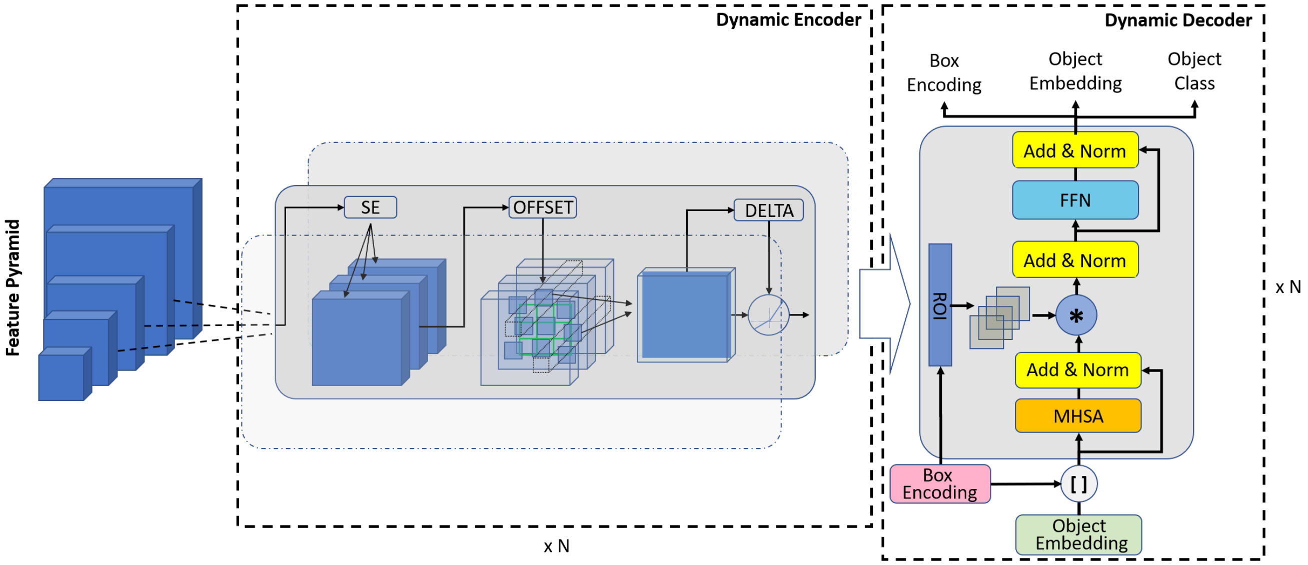

Figure 2. The architecture overview of the proposed approach. Our Dynamic DETR coherently combines a dynamic convolution-based encoder and a dynamic Transformer-based decoder.

3D convolution and Pyramid Convolution [26], we can relax the cross-scale modeling into a few scale-equalizing 2D convolution within neighbours:

$$\begin{gathered}\texttt{PyramidConv}(P_i) = \texttt{Sum}(P_i^*)\\ P_i^* = \{\texttt{Upsample}\left(\texttt{Conv}(P_{i-1})\right), \texttt{Conv}(P_i),\\ \texttt{Downsample}\left(\texttt{Conv}(P_{i+1})\right)\}\end{gathered} \tag{3}$$

However, a naive convolution operation is not able to approximate self-attention in spatial domain due to its small kernel size. Here, we further apply Deformable Convolution [7, 28] to attend kernel learning on sparsely spatial locations, which practically formulates a spatial attention. Note that simply replacing each 2D convolution with 2D deformable convolution may fail to model the spatial attention correctly, as each scale may attend to a different spatial location, leading to conflicts when aggregating features by summation. Thus, we propose to only attend to spatial locations learned from an un-resized central layer and propagate to its resized neighbours:

$$\begin{gathered}P_i^+ = \{\texttt{Upsample}\left(\texttt{DeformConv}(P_{i-1}, s_i)\right),\\ \texttt{DeformConv}(P_i, s_i),\\ \texttt{Downsample}\left(\texttt{DeformConv}(P_{i+1}, s_i)\right)\}\\ s_i = \texttt{Offset}(P_i)\end{gathered} \tag{4}$$

Furthermore, we apply SE [12] on scales to formulate a scale attention to fuse the features:

$$w^{P_i} = \texttt{SE}(P_i^+) \tag{5}$$

Afterwards, we also apply Dynamic Relu [4] to formulate a representation (*i.e.*, channel or feature dimension) attention:

$$\begin{gathered}\texttt{DyReLU}(x_c) = \max(a_c^1 x_c + b_c^1, a_c^2 x_c + b_c^2)\\ a_c^1, b_c^1, a_c^2, b_c^2 = \texttt{Delta}(x_c)\end{gathered} \tag{6}$$

Finally, the multi-scale self-attention in our dynamic encoder is formulated as:

$$\texttt{MultiScaleSelfAttn}(P) = \underset{i=1 \ldots k}{\texttt{Concat}} \left(\texttt{DyReLU}(w^{P_i} P_i^+)\right) \tag{7}$$

Interestingly, our dynamic encoder can be viewed as a sequentially decomposed approximation of full self-attention, similar to [8]. Our approach dynamically adjusts attention based on multiple factors such as scale importance, spatial importance, and representation importance. By stacking multiple modules consecutively together, our dynamic encoder can largely improve the sparseness of feature representation, leading to better detection performance. Compared with Deformable DETR [29], which also applies deformable convolution to extract features, our implementation better approximates the attention learning on extra scale and channel dimensions.

3.3. Dynamic Decoder

To lower the learning difficulty in the cross-attention mechanism in Transformer, we propose to utilize mixed attention blocks instead of the traditional multi-head layers. Inspired by recent progress of ConvBERT [13] in language processing, we use dynamic convolution to replace the cross-attention layer. To further adapt this idea to object detection, we first introduce a widely used RoI Pooling [22] layer into the Transformer decoder, as shown in Figure 2. More specifically, we first replace the position embedding with a box encoding $B \in \mathbb{R}^{q \times 4}$ which was initialized as the size of the full image at the beginning. Given the feature output P_{enc} from our dynamic encoder and the box encoding B, we can pool region features $F \in \mathbb{R}^{q \times r \times r \times d}$ from the feature pyramid:

$$F = \texttt{RoIPool}(P_{enc}, B, r) \tag{8}$$

where r is the pooling size, d is the number of channels of P_{enc}. To accomplish this in the cross-attention mechanism, we require a query embedding $Q \in \mathbb{R}^{q \times d}$ for object queries. It first goes through the multi-head self-attention layer:

$$Q^* = \texttt{MultiHeadSelfAttn}(Q, Q, Q) \tag{9}$$

Then we generate dynamic filters based on Q using a fully-connected layer:

$$W^Q = \texttt{FC}(Q^*) \tag{10}$$

Finally, we can perform cross-attention between queries and region features by applying a 1×1 convolution using dynamic filters W^Q:

$$Q^F = \texttt{Conv}_{1\times1}\left(F, W^Q\right) \tag{11}$$

The attended feature Q^F can be further fed into FFN layers to generate different predictions such as new object embedding $\hat{Q}$, new box encoding $\hat{B}$ and object class $\hat{C}$:

$$\hat{Q} = \texttt{FFN}\left(Q^F\right) \tag{12}$$

$$\hat{B} = \texttt{ReLU}\left(\texttt{LN}\left(\texttt{FC}(\hat{Q})\right)\right) \tag{13}$$

$$\hat{C} = \texttt{Softmax}\left(\texttt{FC}(\hat{Q})\right) \tag{14}$$

By stacking our dynamic decoders sequentially, we implicitly attain a coarse-to-fine refining box encoding from a full image at the early layer to a specific object at the final layer. Such a process greatly reduces the learning difficulty of cross-attention by regulating the model to focus on sparse regions first and then expand to global progressively. Since box encoder also has learnable parameters, it will behave as an anchor generator after training convergence. Our Dynamic Decoder can significantly reduce the training epochs needed.

4. Experiments

4.1. Setup

Dataset. We validate our proposed Dynamic DETR on the challenging MS COCO object detection benchmarks [16] following the widely used common practice. MS COCO dataset contains about 160K images collected from web images on 80 common categorise. The dataset is further split into three subsets: train2017 (118K images), val2017 (5K images) and test2017 (41K images). In all our experiments, we only train on train2017 images without using any extra data. For experiments of ablation studies, we evaluate the performances on val2017 subset. When comparing to state-the-of-art methods, we report the official results returned from the test server on test-dev set. We report the standard mean average precision (mAP) under different IoU thresholds and object scales.

Implementation Detail. We use standard ImageNet [9] pre-trained ResNet and ResNeXt as backbones with FPN [14] to extract feature maps from 5 different scales (from 1/4 to 1/64 of input size). The batch norm statics is frozen in backbone similar to [11]. We then feed these feature maps into our dynamic encoder and decoder. We use pool size 7 in ROI Pooling layer. We set the number of pooled box to be same as the number of bounding box encoding and number of query. The hidden dimension of query embedding is 256. We implement our Dynamic DETR in PyTorch.

Training and Inference. We train our models using both standard 1× (12 epochs) schedule without multi-scale training for all ablation studies and prolong 3× (36 epochs) schedule with multi-scale training for comparison with state-of-the-art to demonstrate our advantages. We use AdamW optimizer [18] in both setups and choose a initial learning rate at $1e^{-4}$ and weight decay at $1e^{-4}$. We step down the learning by a rate of 0.1 twice at 67% and 89% of epochs. All the experiments are trained on a node with 8 V100 GPUs. No other augmentation (such as random crop, mosaic, etc) or optimization tricks (such as EMA, weight normalization) were used during training. During inference, we do not use multi-scale testing neither.

4.2. Ablation Study

We conduct a series of ablation studies to demonstrate the effectiveness of our proposed Dynamic DETR using standard 1× (12 epochs) setup.

Method	AP	AP_{50}	AP_{75}
DETR	15.5	29.4	14.5
+ Dynamic Encoder	24.1	40.9	24.8
Deformable DETR	37.2	55.5	40.5
+ Dynamic Encoder	34.3	52.3	37.4
Dynamic Decoder	**40.2**	**58.6**	**43.4**
+ Dynamic Encoder	**42.9**	**61.1**	**46.2**

Table 1. Ablation study on the effectiveness of each components in our Dynamic DETR on MS COCO validation set using 1x setup.

#	AP	AP_{50}	AP_{75}
Ablation on #Encoder:			
2	41.2	59.4	44.3
4	42.4	60.6	45.7
6	**42.9**	**61.0**	**46.1**
Ablation on #Decoder:			
2	29.2	43.1	31.3
4	40.3	57.2	43.4
6	**42.9**	**61.0**	**46.1**
Ablation on #Head in Decoder:			
2	42.5	60.6	45.7
4	**42.9**	**61.1**	**46.2**
8	42.9	61.0	46.1
Ablation on #Dimension in Decoder:			
512	38.1	55.4	41.0
1024	40.5	57.1	43.2
2048	**42.9**	**61.1**	**46.2**
Ablation on #Query:			
100	41.1	60.0	44.5
300	**42.9**	**61.0**	**46.1**

Table 2. Ablation study on the effectiveness of module stacking in our Dynamic DETR on MS COCO validation set using 1x setup.

Effectiveness of the Components. We start with analyzing the effectiveness of our dynamic encoder and dynamic decoder by swapping components with DETR [2] and Deformable DETR [29]. Shown in Table 1, we first add our dynamic encoder in DETR. It is clear to see that by introducing our dynamic encoder, we largely improve the DETR performance on both small and large objects by 8.6%.

Then we apply our dynamic encoder to replace the encoder within Deformable DETR, we do observe a performance drop by 2.9%. Further investigation reveals that the decoder of Deformable DETR solve the slow convergence problem by introducing an initial attention ”anchor points” to cross-attention layer and loss. It doesn't take the full advantage of feature fusion in feature pyramid and is incompatible with our dynamic encoder design. This motivates us to design our dynamic decoder.

Finally, we demonstrate the performance of our own approach. By using only our dynamic decoder, we have already surpassed others by a large margin while only slightly increase the parameters. When final combine our dynamic encoder and dynamic decoder together, we are able to achieve a state-of-the-art performance. We significantly outperform Deformable DETR by 5.7%.

Analysis on Number of Modules. We continue to investigate the influence by varying different number of modules in our approach, shown in Table 2. We first evaluate the performance influence of using different number of dynamic encoders. It is obvious to see that only using two stacks of our dynamic encoders can already achieve good performance at. Stacking more modules can further improve the performance. This proves the effectiveness of our dynamic encoder.

We then evaluate the performance influence of using different number of dynamic decoders. It is clear to see that use less stacks of our dynamic decoder will significantly hurt the performance. This phenomena is expected as we require an enough number of decoders to effectively formulate a coarse-to-fine learning scheme for cross-attention. This demonstrates the importance of our dynamic decoder.

We further investigate the effects of number of self-attention heads in dynamic decoder. It is interesting to observe that we only require 4 heads, which reduces the number by half compared to others. This further demonstrates the effectiveness of our dynamic decoder.

Finally, we investigate the importance of number of queries. We observe a similar phenomena as Deformable DETR that increasing number of queries will continuously increase the final performance. To conduct fair comparison with Deformable DETR, we also use 300 object queries in further experiments.

4.3. Comparison to State-of-the-Art

We compare the performance of our Dynamic DETR with both of state-of-the-art traditional object detectors and concurrent End to end object detectors.

Comparison to Traditional Object Detectors. Since end to end object detectors are often criticized by the unfair training time, we first compare our Dynamic DETR with traditional methods, such as [22, 11, 15, 24, 27, 5, 20] using a standard 1x schedule without multi-scale training. Shown in Table 3, our proposed Dynamic DETR achieves a new state-of-the-art performance at 42.9 mAP, which outperforms previous best [20] by 1.5 mAP without any bells and whistles in training. We also shows the performance of DETR and Deformable DETR using 1x training schedule and dose find both methods yield underrated performance due to short training epochs. The experiments well demonstrates that our Dynamic DETR achieve a superior learning efficiency and performance.

Convergence Analysis. We further conduct a convergence analysis by prolonging the training to 50 epochs and stepping down the learning rate by a factor 0.1 at multiple training epochs. Shown in Figure 3, we compare our convergence curve with Deformable DETR. It is clear to see that our approach reaching a convergence at around 40 epochs. Interestingly, Deformable DETR doesn't reach

Method	Backbone	Iteration	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$
Faster R-CNN[22]	ResNet-50	1x	37.9	58.8	41.1	22.4	41.1	49.1
Mask R-CNN[11]	ResNet-50	1x	38.6	59.5	42.1	22.5	42.0	49.9
RetinaNet[15]	ResNet-50	1x	37.4	56.7	40.3	23.1	41.6	48.3
FCOS[24]	ResNet-50	1x	38.6	57.2	41.7	23.5	42.8	48.9
ATSS[27]	ResNet-50	1x	39.3	57.5	42.8	24.3	43.3	51.3
RepPoints v2[5]	ResNet-50	1x	41.0	59.9	43.9	23.8	44.8	54.0
BorderDet[20]	ResNet-50	1x	41.4	59.4	44.5	23.6	45.1	54.6
DETR*[20]	ResNet-50	1x	15.5	29.4	14.5	4.3	15.1	26.7
Deformable DETR*[20]	ResNet-50	1x	37.2	55.5	40.5	21.1	40.7	50.5
Dynamic DETR	ResNet-50	1x	**42.9**	**61.0**	**46.3**	**24.6**	**44.9**	**54.4**

Table 3. Compared to SOTA under standard 1× setup using the same backbone on MS COCO validation set. * indicates using multi-scale training.

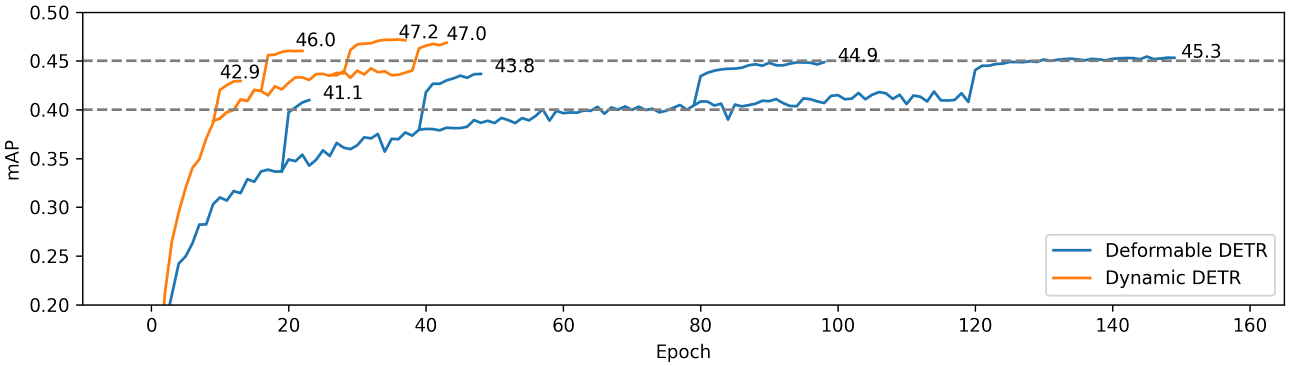

Figure 3. Convergence curves of our Dynamic DETR and Deformable DETR on MS COCO validation set.

convergence at reported 50 epochs, but at a much longer 150 epochs. We can conclude that our Dynamic DETR reduces the training epochs to converge by 4× compared to Deformable DETR, while yield a better performance (47.2 mAP vs 45.3 mAP).

Comparison to SOTA Object Detectors. Finally, we compare to all state-of-the-art object detectors [1, 5, 27, 20, 23, 2, 29] at full convergence on COCO test-dev set. Shown in Table 4, we achieve new state-of-the performances at 47.2 mAP using ResNet-50 backbone and 49.3 using ResNeXt-101-DCN backbone compared to both traditional object detectors and end to end object detectors. The two-stage modification of Deformable DETR (largely improves its performance) can further boost our performance. But we don't include for the cleanliness of presentation.

4.4. Visualization

As mentioned in above section, our dynamic decoder refine the box encoding in a coarse to fine matter to assist the learning of cross-attention layer. To demonstrate that, we visualize the output of updated box encoding after different stage of dynamic decoder layer, shown in Figure 4. We draw the predicted boxes in different colors for better differentiation, not for representing the categories. For the first three rows, we pick images containing objects with significant variants of scale. It is clear to see that, at early stage of decoder, the predicted box cover a large portion of potential region. As the decoder goes deeper, the predicted box becomes more sparse, focusing on dedicated objects. This proves the intention of our design. For the last row, we pick a ”easier” image with different scale of elephants to demonstrate the learning dynamics. At first stage of the decoder, it first learns to attend to the largest elephant with high confidence score. Then as the stage of the decoder goes deeper, it shifts its attention to refine smaller elephants. Again, this example further proves that our dynamic decoder can assist the learning of cross-attention layer refining the box encoding in a coarse to fine matter.

5. Conclusion

In this paper, we address two existing problems of DETR (Detection with Transformers): small feature resolution and

Method	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Cascade-RCNN [1]	ResNet-50	40.6	59.9	44.0	22.6	42.7	52.1
RepPoints v2 [5]	ResNet-50	44.4	63.5	47.7	26.6	47.0	54.6
RepPoints v2 [5]	ResNeXt-101-DCN	49.4	68.9	53.4	30.3	52.1	62.3
ATSS [27]	ResNeXt-101-DCN	47.7	66.5	51.9	29.7	50.8	59.4
BorderDet [20]	ResNeXt-101-DCN	48.0	67.1	52.1	29.4	50.7	60.5
Sparse R-CNN [23]	ResNeXt-101-DCN	48.9	68.3	53.4	29.9	50.9	62.4
DETR[2]	ResNet-50	43.3	63.1	45.9	22.5	47.3	61.1
Deformable DETR [29]	ResNet-50	43.8	62.6	47.7	26.4	47.1	58.0
Deformable DETR (two-stage) [29]	ResNet-50	46.9	66.4	50.8	27.7	49.7	59.9
Deformable DETR (two-stage) [29]	ResNeXt-101-DCN	50.1	69.7	54.6	30.6	52.8	64.7
Dynamic DETR	ResNet-50	**47.2**	**65.9**	**51.1**	**28.6**	**49.3**	**59.1**
Dynamic DETR	ResNeXt-101-DCN	**49.3**	**68.4**	**53.6**	**30.3**	**51.6**	**62.5**

Table 4. Compared to SOTA results using different backbones on MS COCO test-dev set

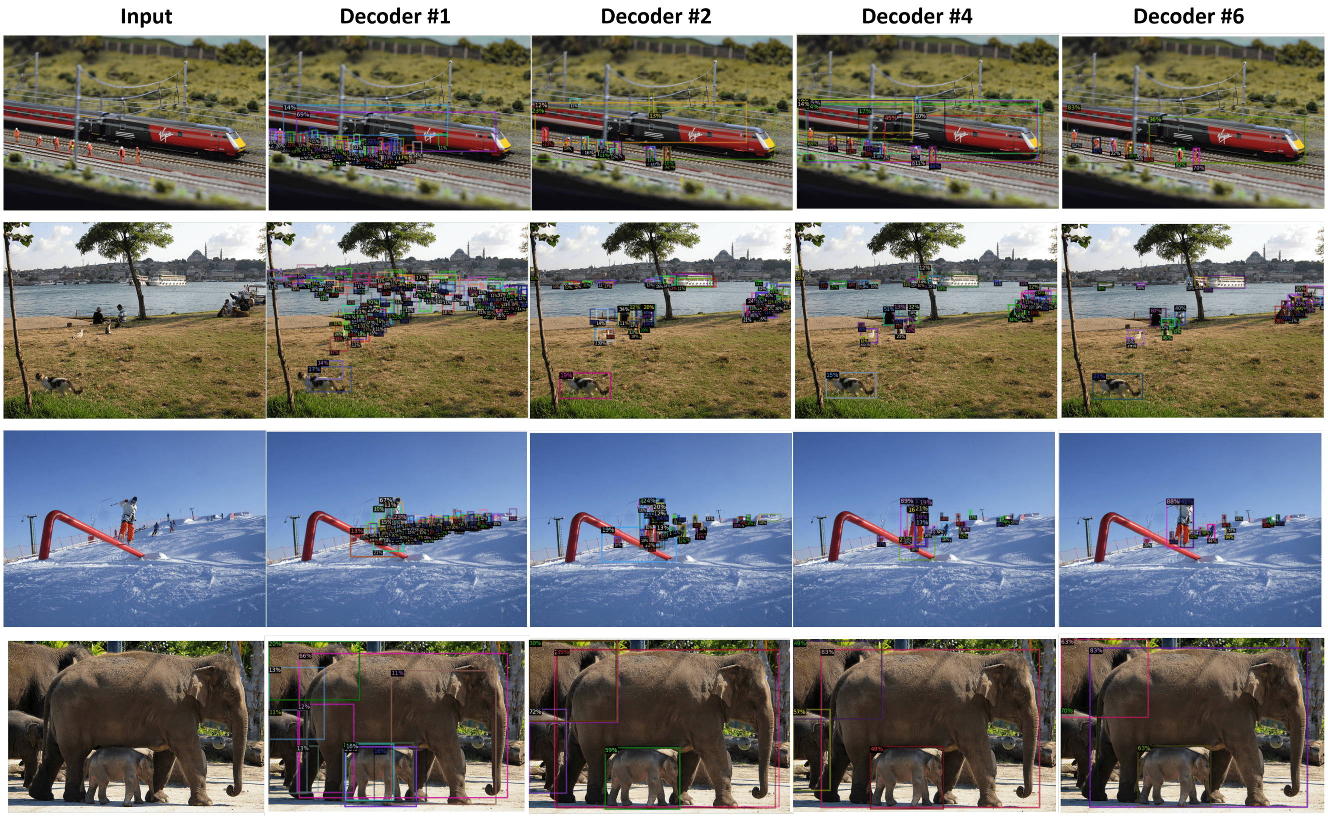

Figure 4. Visualization of box encoding output after different decoder layers. The bounding box is refined in a coarse to fine matter. Best viewed in color and high resolution. The color of box is for better visualization, not stands for classes.

slow training convergence by a novel Dynamic DETR approach. It introduces dynamic attentions into both the encoder and decoder stages. In the encoder stage, we propose to use a convolution-based dynamic encoder with various attention types to approximate the Transformer encoder's attention mechanism. In the decoder stage, we replace the cross-attention module with a ROI-based dynamic attention. Such an approach largely increases the feature resolution and reduces the training epochs needed for convergence. This framework helps end-to-end object detection based on transformers first achieves the best performance with ResNet-50 backbone under 1X training setup.

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. 3, 7, 8

[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. 3, 6, 7, 8

[3] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[4] Y. Chen, X. Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic relu. *ArXiv*, abs/2003.10027, 2020. 2, 4

[5] 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. 6, 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, 3

[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

[8] Xiyang Dai, Yinpeng Chen, Bin Xiao, Dongdong Chen, Mengchen Liu, Lu Yuan, and Lei Zhang. Dynamic head: Unifying object detection heads with attentions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7373–7382, June 2021. 2, 4

[9] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 5

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

[11] 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. 5, 6, 7

[12] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018. 2, 4

[13] Zi-Hang Jiang, Weihao Yu, Daquan Zhou, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. Convbert: Improving bert with span-based dynamic convolution. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12837–12848. Curran Associates, Inc., 2020. 4

[14] 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, 5

[15] 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. 3, 6, 7

[16] 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

[17] 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

[18] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. 5

[19] 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

[20] Han Qiu, Yuchen Ma, Zeming Li, Songtao Liu, and J. Sun. Borderdet: Border feature for dense object detection. In *ECCV*, 2020. 6, 7, 8

[21] 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, 3

[22] 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, 3, 5, 6, 7

[23] 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. 3, 7, 8

[24] 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. 6, 7

[25] 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. 1, 2, 3

[26] 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. 2, 4

[27] 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. 6, 7, 8

[28] 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, 4

[29] 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. 2, 3, 4, 6, 7, 8