Efficient Decoder-free Object Detection with Transformers

Peixian Chen[1†], Mengdan Zhang[1†], Yunhang Shen[1], Kekai Sheng[1], Yuting Gao[1], Xing Sun[1], Ke Li[1⋆], and Chunhua Shen[2]

[1]Tencent Youtu Lab, [2]Zhejiang University
{peixianchen, davinazhang, saulsheng, yutinggao, tristanli}@tencent.com
{shenyunhang01, winfred.sun, chhshen}@gmail.com

Abstract. Vision transformers (ViTs) are changing the landscape of object detection approaches. A natural usage of ViTs in detection is to replace the CNN-based backbone with a transformer-based backbone, which is straightforward and effective, with the price of bringing considerable computation burden for inference. More subtle usage is the DETR family, which eliminates the need for many hand-designed components in object detection but introduces a decoder demanding an extra-long time to converge. As a result, transformer-based object detection can not prevail in large-scale applications. To overcome these issues, we propose a novel decoder-free fully transformer-based (DFFT) object detector, achieving high efficiency in both training and inference stages, for the first time. We simplify objection detection into an encoder-only single-level anchor-based dense prediction problem by centering around two entry points: 1) Eliminate the training-inefficient decoder and leverage two strong encoders to preserve the accuracy of single-level feature map prediction; 2) Explore low-level semantic features for the detection task with limited computational resources. In particular, we design a novel lightweight detection-oriented transformer backbone that efficiently captures low-level features with rich semantics based on a well-conceived ablation study. Extensive experiments on the MS COCO benchmark demonstrate that DFFT$_{\text{SMALL}}$ outperforms DETR by 2.5% AP with 28% computation cost reduction and more than 10× fewer training epochs. Compared with the cutting-edge anchor-based detector RetinaNet, DFFT$_{\text{SMALL}}$ obtains over 5.5% AP gain while cutting down 70% computation cost. The code is available at https://github.com/peixianchen/DFFT.

Keywords: Object detector, Transformers, Efficient network

1 Introduction

Object detection is a classic computer vision task aiming to locate and recognize objects in natural images. Recently, vision transformers [24, 30, 5, 10] have

---

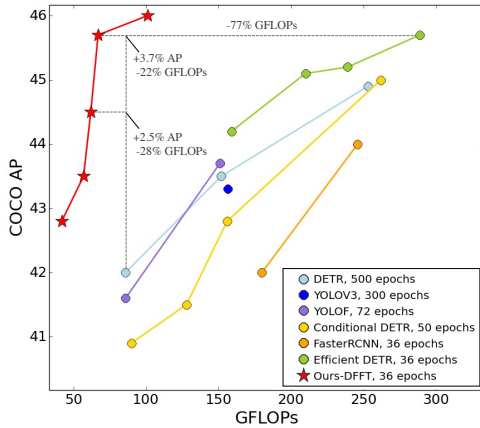⋆ Corresponding author
† Equal Contribution.

Fig. 1: The trade-off between performance (AP) and efficiency (Epochs & GFLOPs) for detection methods. With a lightweight detection-oriented backbone and a decoder-free single-level dense prediction module, DFFT$_{\text{MEDIUM}}$ gets 22% faster inference, more than 10× fewer training epochs and 3.7% higher AP than DETR, cuts down the 77% GFLOPs from Efficient DETR.

been widely developed as powerful backbones in traditional detection frameworks such as Faster RCNN [26], Mask RCNN [17], and RetinaNet [20]. However, these transformer-based detectors achieve high precision at the expense of computational efficiency (*e.g.*, at least 200 GFLOPs), which precludes their use in real-world applications with limited resources. DETR [2] is a pioneering work that addresses this issue by using an encoder-decoder transformer design that reduces object detection to an end-to-end set prediction problem. DETR's novel decoder helps object queries attend to diverse regions of interest on single-level representation, considerably boosting inference efficiency (ranging from 86 to 253 GFLOPs). Unfortunately, this enhancement comes at the expense of around 10× to 20× slower training convergence. As a result, it remains open whether transformer-based detectors can attain high precision without losing efficiency in training and inference stages.

Recent work in the DETR family has mainly focused on improving the delayed convergence induced by the decoder. They augment object queries in the decoder with explicit spatial priors such as reference points [36], anchor points [31, 28], RPN proposals [34, 26], and conditional spatial embeddings [25, 13]. However, introducing spatial priors to the decoder stage sacrifices the detector's inference efficiency, consuming more than 1.5× GFLOPs. It also raises the question whether the above efficient yet accurate transformer-based detector inevitably needs a decoder.

In this paper, we build a novel detection architecture named DFFT: **D**ecoder-**F**ree **F**ully **T**ransformer-based object detector, which achieves both higher accuracy and better training-inference efficiency across a spectrum of low resource constraints (*e.g.*, from 40 to 100 GFLOPs) as shown in Figure 1. Based on a well-conceived analysis of how different transformer architectures (*e.g.*, attention components' type, position and linkage) involved in the backbone, feature fusion, and class/box network, impact the trade-off between detection performance and efficiency, DFFT simplifies the whole object detection pipeline to

an encoder-only single-level anchor-based dense prediction task. Specifically, our design of DFFT centers around two entry points:

Entry Point 1: Eliminate the training-inefficient decoder and leverage two strong encoders to preserve the accuracy of single-level feature map prediction. To design a light-weight detection pipeline comparable to DETR and maintain high training efficiency, we eliminate the training-inefficient decoder and propose two strong transformer encoders in the feature fusion and class/box network to avoid performance decline after trimming the decoder. Benefiting from two strong encoders, DFFT conducts anchor-based dense prediction only on a single-level feature map, ensuring training and inference efficiency while maintaining high accuracy. (1) The *scale-aggregated encoder* summarizes multi-scale cues to one feature map by progressively analyzing global spatial and semantic relations of two consecutive feature maps. Thus, instances of various scales are easily detected on the single feature map, avoiding exhaustive search across network layers. (2) The *task-aligned encoder* enables DFFT to conduct classification and regression simultaneously in a coupled head. By taking advantage of group channel-wise attention, it resolves the learning conflicts from the two tasks and provides consistent predictions [27, 6].

Entry Point 2: Explore low-level semantic features as much as possible for the detection task with limited computational resources. We design a strong and efficient detection-oriented transformer backbone after an in-depth study on different characteristics of transformer attention components (e.g., spatial-wise attention and channel-wise attention). Furthermore, we propose to incorporate semantic-augmented attention modules into several stages of the backbone to capture rich low-level semantics. Low-level semantics from different stages help the detector distinguish distractors in detail. Such design is quite different from common backbones [24, 30, 10] that are dedicated to learn final high-level semantics for the classification task.

Finally, we conduct comprehensive experiments to verify the superiority of DFFT as well as the effectiveness of all the above designs. Compared to the foundation work deformable DETR [36], DFFT achieves 61% inference acceleration, 28% training acceleration, and 1.9% AP gain.

2   Related Work

2.1   One-stage and Two-stage Detectors

Mainstream detectors exploit two types of anchors: anchor boxes [26, 20] and anchor points [29, 9]. Anchors are generated at the center of each sliding-window position to offer candidates for objects. Typical one-stage detectors [4, 9, 29, 15, 20] directly predict categories and offsets of anchors for the whole feature maps, while two-stage detectors [17, 26, 1] first generate region proposals from dense anchor boxes by a Region Proposal Network (RPN) [26] and then refine the detection for each proposed region afterward.

One main challenge in object detection is to represent objects at vastly different scales effectively. Both one-stage and two-stage detectors overcome it with

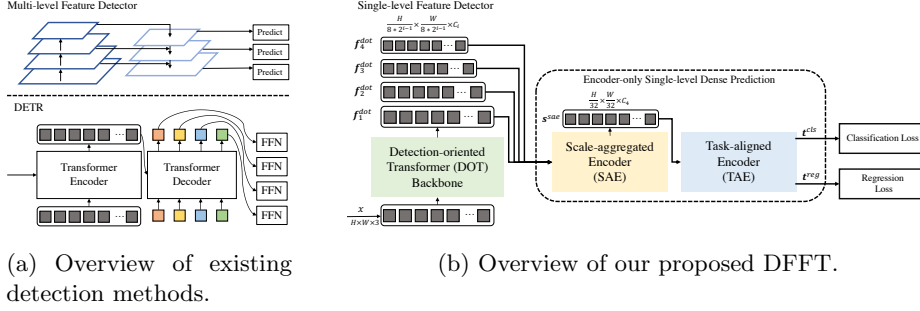(a) Overview of existing detection methods.

(b) Overview of our proposed DFFT.

Fig. 2: Overview of existing detection methods and our proposed Decoder-Free Fully Transformer-based (DFFT) detector. **(a)** Existing methods either rely on multi-level feature detectors [20] or adopt the DETR framework [2]. **(b)** Our DFFT simplifies object detection to an encoder-only single-level dense prediction framework. The proposed two strong encoders enables us to conduct fast but accurate inference on a single-level feature map, outperforming existing multi-level feature detectors. The design also trims the training-inefficient decoder for more than $10\times$ training acceleration over DETR. We also propose a novel lightweight detection-oriented transformer backbone to capture richer low-level semantic features and further boost object detection.

multi-scale features and multi-level predictions. FPN [19] is widely used in these detectors [29, 17, 20], which builds feature pyramid by sequentially combining two adjacent layers in feature hierarchy in backbone model with top-down and lateral connections. Later CNN-based designs on cross-scale connections use bottom-up paths [23], U-shape modules [35], and the neural architecture search [16, 32]. YOLOF [4] provides an alternative solution, which exploits dilated encoder to detect all objects on single-level features. In contrast, our DFFT introduces large receptive fields to cover large objects based on the transformer's global relation modeling, and meanwhile, aggregates low-level semantics through the scale-aggregated encoder. Such designs enable DFFT to achieve superior detection performance.

Recently, transformer-based backbones [5, 30, 24, 10] have shown superior performance in object detection based on standard frameworks such as MaskR-CNN [17, 1] and RetinaNet [20]. However, these backbones are usually directly plugged into the framework without regard for the effects of replacing CNN with transformers. These methods typically consume enormous computation costs (e.g., over 300 GFLOPs for MobileFormer [5]). DFFT is the first method to explore efficient and fully transformer-based detection.

2.2   End-to-end Detectors

End-to-end detectors [2, 36, 11, 25, 13] remove the complicated post-processing like NMS and achieve one-to-one matching between the target and the candi-
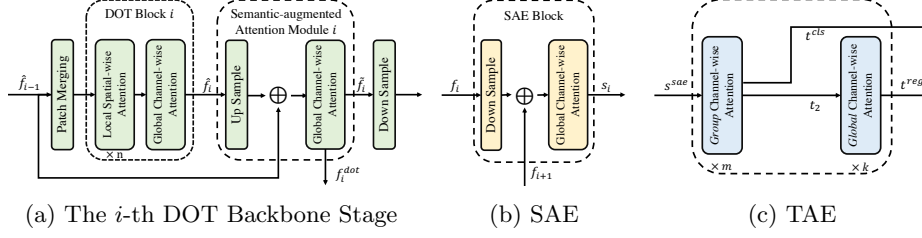
Fig. 3: Illustration of the three major modules in our proposed DFFT. DFFT contains a light-weight **D**etection-**O**riented **T**ransformer backbone with four DOT stages to extract features with rich semantic information, a **S**cale-**A**ggregated **E**ncoder (SAE) with three SAE blocks to aggregate multi-scale features into one feature map for efficiency, and a **T**ask-**A**ligned **E**ncoder (TAE) to resolve conflicts between classification and regression tasks in the coupled detection head.

date by the Hungarian algorithm. DETR [2] uses an encoder-decoder transformer framework. The transformer encoder processes the flattened deep features from the CNN backbone. The non-autoregressive decoder takes the encoder's outputs and a set of learned object query vectors as the input, and predicts the category labels and bounding boxes accordingly. The decoder's cross-attention module attend to different locations in the image for different object queries, which requires high-quality content embeddings and thus training costs. DETR needs a long training process (500 epochs) and is not suitable for small objects. Deformable DETR [36] accelerates the convergence via learnable sparse sampling and multi-scale deformable encoders. It generates a reference point for each object query and uses deformable attention to make each reference point only focus on a small fixed set of sampling points. Anchor DETR [31] exploits anchor points to accelerate training. Conventional designs such as RPN [34], RCNN, and FCOS [28] are also used to optimize the DETR framework. Although better performance and fast convergence are achieved, the computation cost increases significantly (e.g., $2\times$ GFLOPs in deformable DETR) due to multi-scale feature encoding [36, 28]. Moreover, dense priors such as reference points [36], anchor points [31, 28], proposals [28], and conditional spatial embedding [25] are introduced to optimize the DETR pipeline for fast convergence. It shows that DETR is not the only solution for efficient, fully transformer-based detectors. Compared with them, DFFT trims the decoder, also ensures both fast training convergence and inference while maintains comparable performance.

3 Method

In this section, we introduce DFFT, an efficient Decoder-Free Fully Transformer-based object detector. An overview of DFFT is illustrated in Figure 2b. The Detection-oriented Transformer backbone F extracts features at four scales and sends them to the following encoder-only single-level dense prediction module.

The prediction module first aggregates the multi-scale feature into a single feature map through the Scale-Aggregated Encoder S. Then we use the Task-Aligned Encoder T to align the feature for classification and regression tasks simultaneously for higher inference efficiency.

3.1   Detection-oriented Transformer Backbone

Detection-oriented transformer (DOT) backbone aims to extract multi-scale features with strong semantics. As shown in Figure 3a, it hierarchically stacks one embedding module and four DOT stages, where a novel semantic-augmented attention module aggregates the low-level semantic information of every two consecutive DOT stages. For each input image $\boldsymbol{x} \in \mathbb{R}^{H \times W \times 3}$, the DOT backbone extracts features at four different scales:

$$\boldsymbol{f}_1^{\mathrm{dot}}, \boldsymbol{f}_2^{\mathrm{dot}}, \boldsymbol{f}_3^{\mathrm{dot}}, \boldsymbol{f}_4^{\mathrm{dot}} = F(\boldsymbol{x}), \tag{1}$$

where $\boldsymbol{f}_i^{\mathrm{dot}} \in \mathbb{R}^{\frac{H}{8 \cdot 2^{i-1}} \times \frac{W}{8 \cdot 2^{i-1}} \times C_i}$ is the i-th feature with C_i channels for $i \in \{1, 2, 3, 4\}$. In what follows, we expand the formalization of the DOT backbone function F.

Embedding Module. For an input image $\boldsymbol{x} \in \mathbb{R}^{H \times W \times 3}$, we first divide it into $\frac{H \times W}{8 \times 8}$ patches and feed these patches to a linear projection to obtain patch embeddings $\hat{\boldsymbol{f}}_0$ of size $\frac{H}{8} \times \frac{W}{8} \times C_1$, written as

$$\hat{\boldsymbol{f}}_0 = F_{\mathrm{embed}}(\boldsymbol{x}) \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times C_1}, \tag{2}$$

where F_{embed} is the embedding module described above.

DOT Block. Each DOT stage contains one DOT block F_{block}, designed to efficiently capture both the local spatial and the global semantic relations at each scale. When processing high-resolution feature maps in dense prediction, conventional transformer blocks reduce computational costs by replacing the multi-head self-attention (MSA) layer with the local spatial-wise attention layer, such as spatial-reduction attention (SRA) [30] and shifted window-based multi-head self-attention (SW-MSA) [24]. However, this design sacrifices detection performance as it only extracts multi-scale features with limited low-level semantics.

To mitigate this shortcoming, our DOT block includes multiple SW-MSA blocks [24] and one global channel-wise attention block [10], as illustrated in the first part of Figure 3a. Note that each attention block contains an attention layer and an FFN layer, and we omit the FFN layer in each attention block of Figure 2 to simplify the illustration. We denote by $\hat{\boldsymbol{f}}_i$ the DOT block's output feature at the i-th DOT stage. We find that placing a lightweight channel-wise attention layer behind consecutive local spatial-wise attention layers can benefit deducing object semantics at each scale.

Semantic-Augmented Attention. While the DOT block has enhanced the semantic information in low-level features through the global channel-wise attention, semantics can be improved even further to benefit the detection task. Thus, we propose a novel *semantic-augmented attention* (SAA) module $F_{\mathrm{se\text{-}att}}$,

which exchanges semantic information between two consecutive scale levels and augments their features. SAA consists of an up-sampling layer and a global channel-wise attention block. We incorporate SAA into every two consecutive DOT blocks, as illustrated in the second part of Figure 3a. Formally, SAA takes the outputs from the current DOT block and the former DOT stage, and then returns the semantic augmented feature, which is sent to the next DOT stage and also contributes to the final multi-scale feature $\boldsymbol{f}_i^{\text{dot}}$. We denote by $\widehat{\boldsymbol{f}}_i$ SAA's output feature at the i-th DOT stage.

DOT Stage. The final DOT backbone contains four DOT stages F_{stage} where each stage consists of one DOT block and one SAA module (except the first stage). Specifically, the first stage contains one DOT block and no SAA module, because the inputs of the SAA module are from two consecutive DOT stages. Each of the remaining three stages contains a patch merging module to reduce the number of patches similar to [24], a DOT block, and a SAA module followed by a down-sampling layer to recover the input dimension, as shown in Figure 3a. Thus, the formulation of the DOT block in the i-th stage can be defined as

$$\hat{\boldsymbol{f}}_i = \begin{cases} F_{\text{block}}(\hat{\boldsymbol{f}}_{i-1}), & i = 1, 2 \\ F_{\text{block}}(\text{down}(\widetilde{\boldsymbol{f}}_{i-1})), & i = 3, 4 \end{cases} \tag{3}$$

where down denotes the downsampling function.

The i-th stage's SAA module can be defined as

$$\widetilde{\boldsymbol{f}}_i = \begin{cases} F_{\text{se-att}}\left(\text{up}(\hat{\boldsymbol{f}}_i) + \hat{\boldsymbol{f}}_{i-1}\right), & i = 2 \\ F_{\text{se-att}}\left(\text{up}(\hat{\boldsymbol{f}}_i) + \widetilde{\boldsymbol{f}}_{i-1}\right), & i = 3, 4 \end{cases} \tag{4}$$

where up denotes the upsampling function.

The final multi-scale feature from the DOT backbone can be written as

$$\boldsymbol{f}_i^{\text{dot}} = \begin{cases} \widetilde{\boldsymbol{f}}_{i+1}, & i = 1, 2, 3 \\ \hat{\boldsymbol{f}}_i, & i = 4 \end{cases} \tag{5}$$

3.2   Encoder-only Single-level Dense Prediction

This module is designed to improve both the inference and training efficiency of the fully transformer-based object detector with two novel encoders. It first uses the *scale-aggregated* encoder (SAE) to aggregate the multi-scale features $\boldsymbol{f}_i^{\text{dot}}$ from the DOT backbone into one feature map $\boldsymbol{s}^{\text{sae}}$. After that, it uses the *task-aligned* encoder (TAE) to generate aligned classification feature $\boldsymbol{t}^{\text{cls}}$ and regression feature $\boldsymbol{t}^{\text{reg}}$ simultaneously in a single head.

Scale-aggregated Encoder. We design this encoder with three SAE blocks, as illustrated in Figure 3b. Each SAE block takes two features as the input and aggregates the features step by step across all SAE blocks. We set the scale of final aggregated feature to $\frac{H}{32} \times \frac{W}{32}$ to balance the detection precision and

computational costs. For this purpose, the last SAE block will up-sample the input feature to $\frac{H}{32} \times \frac{W}{32}$ before aggregation. This procedure can be described as

$$
\begin{aligned}
\boldsymbol{s}_0 &= \boldsymbol{f}_1^{\mathrm{dot}}, \\
\boldsymbol{s}_1 &= S_{\mathrm{att}}\big(\mathrm{down}(\boldsymbol{s}_0) + \boldsymbol{f}_2^{\mathrm{dot}}\big), \\
\boldsymbol{s}_2 &= S_{\mathrm{att}}\big(\mathrm{down}(\boldsymbol{s}_1) + \boldsymbol{f}_3^{\mathrm{dot}}\big), \\
\boldsymbol{s}_3 &= S_{\mathrm{att}}\big(\boldsymbol{s}_2 + \mathrm{up}(\boldsymbol{f}_4^{\mathrm{dot}})\big),
\end{aligned}
\tag{6}
$$

where S_{att} is the global channel-wise attention block and $\boldsymbol{s}^{\mathrm{sae}} = \boldsymbol{s}_3$ is the final aggregated feature map.

Task-aligned Encoder. Recent one-stage detectors [4, 29] perform object classification and localization independently with two separate branches (e.g., decoupled head). This two-branch design omits the interaction between two tasks and leads to inconsistent predictions [12, 14]. Meanwhile, feature learning for two tasks in a coupled head usually exists conflicts [7, 27]. We propose the task-aligned encoder which offers a better balance between learning task-interactive and task-specific features via stacking *group* channel-wise attention blocks in a coupled head.

As shown in Figure 3c, this encoder consists of two kinds of channel-wise attention blocks. First, the stacked *group* channel-wise attention blocks T_{group} align and finally split the aggregated feature $\boldsymbol{s}^{\mathrm{sae}}$ into two parts. Second, the *global* channel-wise attention blocks T_{global} further encode one of the two split features for the subsequent regression task. This procedure can be described as

$$
\begin{aligned}
\boldsymbol{t}_1, \boldsymbol{t}_2 &= T_{\mathrm{group}}(\boldsymbol{s}^{\mathrm{sae}}), \\
\boldsymbol{t}^{\mathrm{cls}} &= \boldsymbol{t}_1, \\
\boldsymbol{t}^{\mathrm{reg}} &= T_{\mathrm{global}}(\boldsymbol{t}_2),
\end{aligned}
\tag{7}
$$

where $\boldsymbol{t}_1, \boldsymbol{t}_2 \in \mathbb{R}^{\frac{H}{32} \times \frac{W}{32} \times 256}$ are the split features, and $\boldsymbol{t}^{\mathrm{cls}} \in \mathbb{R}^{\frac{H}{32} \times \frac{W}{32} \times 256}$ and $\boldsymbol{t}^{\mathrm{reg}} \in \mathbb{R}^{\frac{H}{32} \times \frac{W}{32} \times 512}$ are the final features for the classification and regression tasks, respectively.

Specifically, the differences between the *group* channel-wise attention block and the *global* channel-wise attention block lie in that all the linear projections except the projections for key/query/value embeddings in the *group* channel-wise attention block are conducted in two groups. Thus, features interact in attention operations while deduced separately in output projections.

3.3   Miscellaneous

Since DFFT conducts the single-level dense prediction on a single feature map, the pre-defined anchors are sparse. Applying the Max-IoU matching [20] based on the sparse anchors will cause an imbalance problem for positive anchors, making detectors pay attention to large ground-truth boxes while ignoring the small ones when training. To overcome this problem, we use the uniform matching strategy

Table 1: The definition and performance of DFFT models with different magnitudes. In the backbone setting, we list the output feature's number of channels C_i and the number of SA blocks in all four backbone stages. In the effectiveness evaluation, we report the accuracy of the pre-trained backbone on ImageNet and the detection AP of DFFT after training on the MS COCO dataset.

Models	Backbone Settings		Effectiveness (%)		Efficiency (GFLOPs)	
	Value of C_i	Number of SA	Accuracy	AP	Backbone	DFFT
$DFFT_{NANO}$	$(3, 3, 6, 9)$	$(2, 2, 6, 2)$	80.0	42.8	26	42
$DFFT_{TINY}$	$(4, 4, 8, 12)$	$(1, 1, 5, 1)$	81.1	43.5	39	57
$DFFT_{SMALL}$	$(4, 4, 8, 12)$	$(2, 2, 6, 2)$	82.1	44.5	44	62
$DFFT_{MEDIUM}$	$(4, 4, 7, 12)$	$(2, 2, 18, 2)$	82.7	45.7	48	67
$DFFT_{LARGE}$	$(6, 6, 8, 12)$	$(2, 2, 18, 2)$	83.1	46.0	83	101

proposed by YOLOF [4] to ensure that all ground-truth boxes uniformly match with the same number of positive anchors regardless of their sizes. Similar to the setting of most conventional detection methods [29, 4, 20], our loss function consists of a focal loss for classification and a generalized IOU loss for regression. At the inference stage, we conduct object detection efficiently based on the final aggregated feature map s^{sae} with a single pass.

4 Experiments

We evaluate our proposed DFFT on the challenging MS COCO benchmark [21] following the commonly used setting. It contains around 160K images of 80 categories. We compare DFFT with conventional one-stage/two-stage detection methods and DETR-based methods. We also provide a comprehensive ablation study to quantitatively analyze the effectiveness of each module in DFFT. The standard mean average precision (AP) metric is used to measure detection under different IoU thresholds and object scales.

4.1 Settings

The DOT backbone is pre-trained on ImageNet [8] with the same setting as [24]. We train DFFT with the standard $1\times$ (12 epochs) and $3\times$ (36 epochs) training configurations as introduced in [24]. We use the AdamW [18] optimizer with a batch size of 32, an initial learning rate of $1e - 4$ and weight decay of 0.05. The learning rate is stepped down by a factor of 0.1 at the 67% and 89% of training epochs. We conduct all experiments on 8 V100 GPUs.

We implemented models with different magnitudes. The settings and performance of these backbones are shown in Table 1, where C_i denotes the number of channels of the i-th DOT stage's output feature, and the number of SA blocks within each DOT stage is also provided. Only one global channel-wise attention block is added to the end of each stage. For each model, accuracy refers to the backbone's accuracy on ImageNet and AP refers to the precision after training on the MS COCO dataset. All GFLOPs are obtained on the MS COCO dataset.

Table 2: Comparison of our DFFT and modern detection methods on the MS COCO benchmark [21]. The table is divided into four sections from top to bottom: (1) anchor-based methods, (2) DFFT trained for 12 epochs, (3) DETR-based methods, and (4) DFFT trained for 36 epochs. DFFT achieves competitive precision with significantly fewer training epochs and inference GFLOPs.

Methods	Epochs	AP (%)	AP_{50} (%)	AP_{75} (%)	AP_S (%)	AP_M (%)	AP_L (%)	GFLOPs
Faster RCNN-FPN-R50 [26]	36	40.2	61.0	43.8	24.2	43.5	52.0	180
RetinaNet [20]	12	35.9	55.7	38.5	19.4	39.5	48.2	201
YOLOF-R50 [4]	12	37.7	56.9	40.6	19.1	42.5	53.2	86
Swin-Tiny-RetinaNet [24]	12	42.0	-	-	-	-	-	245
Focal-Tiny-RetinaNet [33]	12	43.7	-	-	-	-	-	265
Mobile-Former [5]	12	34.2	53.4	36.0	19.9	36.8	45.3	322
$DFFT_{NANO}$	12	39.1	58.3	41.7	19.0	42.9	51.2	42
$DFFT_{SMALL}$	12	41.4	60.9	44.5	20.1	45.4	58.9	62
$DFFT_{MEDIUM}$	12	42.6	62.5	45.5	22.6	46.7	61.4	67
DETR-R50 [2]	500	42.0	62.4	44.2	20.5	45.8	61.1	86
WB-DETR [22]	500	39.6	58.4	43.8	18.2	42.7	54.9	62
YOLOS [11].	150	37.6	-	-	-	-	-	172
Deformable DETR [36]	50	43.8	62.6	47.7	26.4	47.1	58.0	173
SMCA-R50 [13]	50	43.7	63.6	47.2	24.2	47.0	60.4	152
Anchor DETR-DC5-R50 [31]	50	44.2	64.7	47.5	24.7	48.2	60.6	151
Conditional DETR-R50 [25]	50	40.9	61.8	43.3	20.8	44.6	59.2	90
TSP-FCOS-R50 [28]	36	43.1	62.3	47.0	26.6	46.8	55.9	189
Efficient DETR-R50 [34]	36	44.2	62.2	48.0	**28.4**	47.5	56.6	159
$DFFT_{NANO}$	36	42.8	61.9	46.2	23.4	46.8	59.7	42
$DFFT_{SMALL}$	36	44.5	63.6	48.0	24.5	49.0	60.7	62
$DFFT_{MEDIUM}$	36	**45.7**	**64.8**	**49.7**	25.5	**50.4**	**63.1**	67

4.2   Main Results

Compare with two-stage/one-stage detection methods. The performance of conventional two-stage/one-stage detection methods is shown in the first part of Table 2. Overall, anchor-based methods converge fast within only 12 epochs, and the transformer-based methods generally outperform CNN-based methods. For instance, Focal-Tiny-RetinaNet [33] achieves 7.8% higher AP than the original RetinaNet [20]. However, such good performance comes at the expense of high computational costs; most of these methods need 170 GFLOPs at the minimum. Even the more efficient single-level feature detection method YOLOF [4] needs 86 GFLOPs when using ResNet-50 as the backbone.

The performance of our proposed DFFT with 12 epochs is shown in the second part of Table 2. In contrast to the above methods that endure an obvious trade-off between detection precision and inference efficiency, our DFFT can improve these two metrics simultaneously. For example, $DFFT_{NANO}$ decreases 51% GFLOPs of YOLOF while still increasing 1.4% AP, and $DFFT_{MEDIUM}$ achieves 42.6% AP with only 67 GFLOPs.Furthermore, DFFT reduces 200 GFLOPs from the best-performed Focal-Tiny-RetinaNet [33] at the cost of merely 1% lower AP. These comparisons indicate that our DFFT can effectively reduce the computational cost of the inference stage without sacrificing the detection precision.
Compare with DETR-based methods. The performance of DETR-based methods is shown in the third part of Table 2. We observe that DETR-based

methods can achieve better performance and inference efficiency but converges slower. For instance, DETR only needs 86 GFLOPs at the inference stage to achieve 42.0% AP, but it requires as large as 500 epochs to converge. WB-DETR [22] can only achieve 39.6% AP if given the same training epochs. Subsequent optimized DETR-based methods improve the convergence speed but at the cost of inference efficiency. For example, while Deformable DETR [36] and Condition DETR [25] need 50 epochs to converge and TSP-FOCS [28] and Efficient DETR [34] need only 36 epochs to converge, their GFLOPs are around 4%–120% larger than DETR.

The performance of our DFFT with 36 epochs is listed in the fourth part of Table 2. In contrast to the DETR-based methods that endure a hard-to-optimize trade-off between the convergence and inference efficiency, our DFFT can achieve state-of-the-art detection prevision without sacrificing neither of these two metrics. Compared with DETR [36], our DFFT$_{NANO}$ model improves $13\times$ convergence speed and decreases 51% GFLOPs while achieving significant detection precision (42.0% vs. 42.8%). Compared with Efficient DETR [34] under the same number of training epochs, our model achieves state-of-the-art 45.7% AP with only 67 GFLOPs (57% lower). We further demonstrate the convergence curves of DFFT and DETR-based methods in Figure 4. DFFT reduces 28%–92% training epochs of state-of-the-art methods.
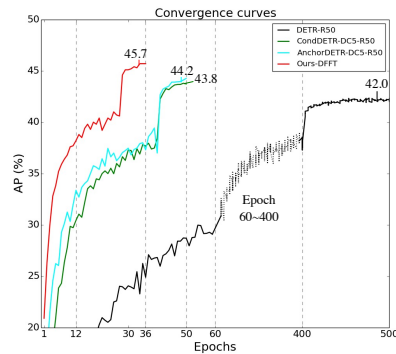

Fig. 4: The convergence curves of DFFT and DETR-based methods on the COCO 2017 validation set. Our DFFT converges significantly faster than the counterparts.

These comparisons verify that our DFFT can effectively optimize training and inference efficiency while achieving competitive and even state-of-the-art detection precision.

4.3 Ablation Study

We provide a comprehensive ablation study of all designs in our DFFT: the detection-oriented transformer (DOT) backbone, scale-aggregated encoder (SAE), and task-aligned encoder (TAE). The ablation study starts with the major components in DFFT, followed by the specific design of each component. We conduct all the experiments on DFFT$_{SMALL}$ model that is trained for 12 epochs.

Major Components. We first evaluate the efficacy of the major components in DFFT. We disable each component by replacing it with a vanilla method as they are not easily removable from our detection framework. Specifically, we (1) replace the DOT backbone (line 1) with Swin-Transformer [24] with the similar GFLOPs; (2) disable the SAE module (lines 1, 2, 4) by directly upsampling the

last stage's outputs to $\frac{H}{32} \times \frac{W}{32}$ and feed them to TAE; and (3) replace the TAE module (lines 1–3) with YOLOF's head. The results are shown in Table 3.

Firstly, the DOT backbone promotes the precision from 33.8% to 37.9%, indicating that it can obtain better semantic features that are more suitable for the detection task. Even without the SAE, it gets competitive precisions with only the last stage's outputs. This also suggests that the SAA can capture multi-scale information when aggregating semantic information.

Table 3: Ablation study of the three major modules in DFFT.

DOT	SAE	TAE	AP (%)	GFLOPs
-	-	-	33.8	45
✔	-	-	37.9	47
✔	✔	-	39.9	58
✔	-	✔	39.8	51
✔	✔	✔	41.4	62

Secondly, SAE further improves the precision to 39.9% by aggregating multi-scale features into one feature map. Yet, disabling SAE would decrease the precision by 1.6%.

Finally, disabling TAE would decrease the prevision by 1.5%. It verifies the necessity of using TAE to align and encode both the classification and regression features.

Detection-oriented Transformer (DOT) block. Table 4 studies how the global channel-wise attention (GCA) and semantic-augmented attention (SAA) contributes to DOT's performance. We only modify the backbone network without disabling the SAE and TAE modules. Lines 1 and 2 show that switching from SW-MSA to our global channel-wise attention can improve 1.1% precision without significant impact on FLOPs. Lines 2 and 3 show that adding one GCA block costs 11 GFLOPs yet only gains 0.3% AP. Once we replace the GCA block with the SAA module (so

Table 4: Ablation study of the DOT backbone. GCA means adding a global channel-wise attention at the end of each DOT block. In the third line, we replace SAA with GCA.

GCA	SAA	AP (%)	GFLOPs
-	-	39.0	48.59
✔	-	40.1	48.97
✔	GCA	40.4	58.65
✔	✔	41.4	62.23

that the two settings have the same number of attention nodes), the precision increases from 40.1% to 41.4%. These two observations suggest that SAA can enhance performance, and having more attention nodes is not the primary cause.

We further visualize the feature maps from each backbone stage of DFFT$_{\text{MEDIUM}}$ trained 12 epochs in Figure 5. In the first two stages, our DOT backbone can obtain low-level features with sufficient semantic information to capture small objects. The third stage then focuses on medium and large objects. Finally, the last stage only responds to large objects. These observations verify that our DOT backbone can enhance semantic information in low-level features and thus boosting the detection precision.

Semantic-Augmented Attention (SAA). SAA obtains richer low-level semantic features for object detection task by augmenting the semantic information from high-level features to the low-level ones, sharing a similar effect as
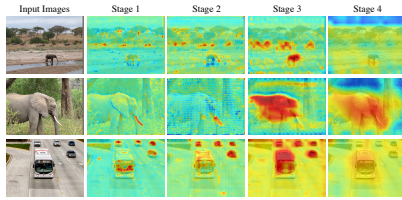
Fig. 5: Visualization of the feature map obtained by each DOT backbone stage. The first two stages focus on small objects while the third stage focuses on medium or larger objects. The last stage only responses to large objects.
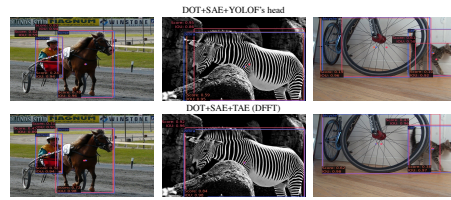
Fig. 6: Illustration of detection results from the best anchors for classification (red) and localization (orange). Ground-truth is indicated by blue boxes and centers. Our TAE helps provide consistency in predictions of classification and localization.

FPN [19]. For a fair comparison with FPN, we disable the two encoders and directly feed features from the backbone to RetinaNet's head, which is a multi-level feature head that accepts four-scale features. The results are shown in Table 5. While both SAA and FPN improves the precision, SAA obtains 0.5% higher AP with 9 fewer GFLOPs than FPN. Thus, the global channel-wise attention suits the transformer better than FPN. Including SAA within the forward pass can obtain an even stronger model.

Scale-aggregated Encoder (SAE). SAE aggregates multi-scale features into one feature map to reduce the inference stage's computational costs. We compare SAE with a similar design in YOLOF, which exploits a dilated encoder to convert features from multiple scales. Table 6 shows that SAE improves 1.1% AP from the dilated encoder of YOLOF. When compared with a vanilla concatenation operation, SAE gets 1.8% higher precision. Overall, SAE can achieve better performance with low computational costs.

Task-aligned Encoder (TAE). Benefiting from the group channel-wise attention's capability of modeling semantic relations, TAE handles task conflicts in a coupled head and further generates task-aligned predictions in a single pass. As shown in the first row of Figure 6, after replacing TAE with YOLOF's head in the baseline model, the best anchors for classification (red) and localization (orange) are distant from each other. That is because YOLOF [4] uses a task-unaligned decoupled head that leads to inconsistent predictions of classification and localization. Comparatively, our TAE provides aligned predictions with both high classification and IOU scores (*e.g.*, person, zebra and cat in Figure 6).

Analysis of the impact on GFLOPs and FPS. We compare the performance of different models in terms of prediction AP and inference GFLOPs and FPS in Table 7. For computational performance, GFLOPs and FPS are sensitive to the number of channels and the number of attention blocks, respectively. For example, compared with $DFFT_{NANO}$, $DFFT_{TINY}$ increases 15 GFLOPs but has a better FPS due to fewer attention blocks, and $DFFT_{SMALL}$ increases 20

Table 5: Analysis of SAA. We disable the two encoders and uses the RetinaNet's head, which is a multi-level feature head accepting four-scale features.

SAA	FPN	AP (%)	GFLOPs
-	-	37.4	319
✔	-	38.9	332
-	✔	38.4	341

Table 6: Analysis of SAE. CONCAT: the direct concatenation of multiple features. YOLOF: exploiting a dilated encoder to convert features.

Method	AP (%)	GFLOPs
CONCAT	39.6	56
YOLOF	40.3	58
DFFT	41.4	62

Table 7: Analysis of how the output feature's number of channels C_i and the number of attention blocks impact GFLOPs and FPS. All the results are measured on the same machine with a V100 GPU using mmdetection [3].

Models	Backbone Settings		AP	GFLOPs	FPS
	Value of C_i	Number of SA			
DETR-R50	-	-	42.0	86	24
Deformable DETR	-	-	43.8	173	14
DFFT$_{NANO}$	$(3,3,6,9)$	$(3,3,7,3)$	42.8	42	22
DFFT$_{TINY}$	$(4,4,8,12)$	$(2,2,6,2)$	43.5	57	24
DFFT$_{SMALL}$	$(4,4,8,12)$	$(3,3,7,3)$	44.5	62	22
DFFT$_{MEDIUM}$	$(4,4,7,12)$	$(3,3,19,3)$	45.7	67	17
DFFT$_{LARGE}$	$(6,6,8,12)$	$(3,3,19,3)$	46.0	101	17

GFLOPs but gets a similar FPS. Overall, we achieve better AP and inference efficiency than deformable DETR; at the same FPS, our DFFT has better accuracy and GFLOPS than DETR. Lastly, although we designed the network architecture mainly to reduce GFLOPs, we note that the above observations can also be used to redesign our network and optimize for FPS in the future.

5   Conclusion

In this work, we discover a trade-off between training and inference efficiency that hinders transformer-based object detection in large-scale applications. Rather than porting transformers directly to the conventional framework or optimizing the DETR framework, we propose DFFT, a novel design of fully transformer-based object detectors. It enables efficiency in both the training and inference stages for the first time without sacrificing noticeable detection precision. Extensive evaluation reveals our DFFT's unique advantages in capturing low-level semantic features in object detection, as well as its ability to preserve detection precision while trimming the training-inefficient decoders in DETR. Finally, DFFT achieves state-of-the-art performance while using only half the GFLOPs of previous approaches, indicating a promising future work on the large-scale application of transformers in object detection.

References

1. Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6154–6162 (2018)
2. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision. pp. 213–229. Springer (2020)
3. Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C.C., Lin, D.: MMDetection: Open mmlab detection toolbox and benchmark. arXiv (2019)
4. Chen, Q., Wang, Y., Yang, T., Zhang, X., Cheng, J., Sun, J.: You only look one-level feature. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13039–13048 (2021)
5. Chen, Y., Dai, X., Chen, D., Liu, M., Dong, X., Yuan, L., Liu, Z.: Mobile-former: Bridging mobilenet and transformer. arXiv (2021)
6. Cheng, B., Wei, Y., Shi, H., Feris, R., Xiong, J., Huang, T.: Revisiting rcnn: On awakening the classification power of faster rcnn. In: Proceedings of the European conference on computer vision (ECCV). pp. 453–468 (2018)
7. Dai, X., Chen, Y., Xiao, B., Chen, D., Liu, M., Yuan, L., Zhang, L.: Dynamic head: Unifying object detection heads with attentions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7373–7382 (2021)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
9. Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., Tian, Q.: Centernet: Keypoint triplets for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6569–6578 (2019)
10. El-Nouby, A., Touvron, H., Caron, M., Bojanowski, P., Douze, M., Joulin, A., Laptev, I., Neverova, N., Synnaeve, G., Verbeek, J., et al.: Xcit: Cross-covariance image transformers. arXiv (2021)
11. Fang, Y., Liao, B., Wang, X., Fang, J., Qi, J., Wu, R., Niu, J., Liu, W.: You only look at one sequence: Rethinking transformer in vision through object detection. arXiv (2021)
12. Feng, C., Zhong, Y., Gao, Y., Scott, M.R., Huang, W.: Tood: Task-aligned one-stage object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3510–3519 (2021)
13. Gao, P., Zheng, M., Wang, X., Dai, J., Li, H.: Fast convergence of detr with spatially modulated co-attention. arXiv (2021)
14. Gao, Z., Wang, L., Wu, G.: Mutual supervision for dense object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3641–3650 (2021)
15. Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: Yolox: Exceeding yolo series in 2021. arXiv (2021)
16. Ghiasi, G., Lin, T.Y., Le, Q.V.: Nas-fpn: Learning scalable feature pyramid architecture for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7036–7045 (2019)
17. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)

18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv (2014)
19. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
20. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
21. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
22. Liu, F., Wei, H., Zhao, W., Li, G., Peng, J., Li, Z.: Wb-detr: Transformer-based detector without backbone. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2979–2987 (2021)
23. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8759–8768 (2018)
24. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. arXiv (2021)
25. Meng, D., Chen, X., Fan, Z., Zeng, G., Li, H., Yuan, Y., Sun, L., Wang, J.: Conditional detr for fast training convergence. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3651–3660 (2021)
26. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28**, 91–99 (2015)
27. Song, G., Liu, Y., Wang, X.: Revisiting the sibling head in object detector. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11563–11572 (2020)
28. Sun, Z., Cao, S., Yang, Y., Kitani, K.M.: Rethinking transformer-based set prediction for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3611–3620 (2021)
29. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9627–9636 (2019)
30. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. arXiv (2021)
31. Wang, Y., Zhang, X., Yang, T., Sun, J.: Anchor detr: Query design for transformer-based detector. arXiv (2021)
32. Xu, H., Yao, L., Zhang, W., Liang, X., Li, Z.: Auto-fpn: Automatic network architecture adaptation for object detection beyond classification. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6649–6658 (2019)
33. Yang, J., Li, C., Zhang, P., Dai, X., Xiao, B., Yuan, L., Gao, J.: Focal self-attention for local-global interactions in vision transformers. arXiv (2021)
34. Yao, Z., Ai, J., Li, B., Zhang, C.: Efficient detr: Improving end-to-end object detector with dense prior. arXiv (2021)
35. Zhao, Q., Sheng, T., Wang, Y., Tang, Z., Chen, Y., Cai, L., Ling, H.: M2det: A single-shot object detector based on multi-level feature pyramid network. In: Proceedings of the AAAI. vol. 33, pp. 9259–9266 (2019)
36. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv (2020)