Towards Efficient Use of Multi-Scale Features in Transformer-Based Object Detectors

Gongjie Zhang[†1,2] Zhipeng Luo[†1,3] Zichen Tian[1] Jingyi Zhang[1] Xiaoqin Zhang[4] Shijian Lu[*1]

[1]S-Lab, Nanyang Technological University [2]Black Sesame Technologies [3]SenseTime Research [4]Wenzhou University

gjz@ieee.org zhipeng001@e.ntu.edu.sg shijian.lu@ntu.edu.sg

Abstract

Multi-scale features have been proven highly effective for object detection but often come with huge and even prohibitive extra computation costs, especially for the recent Transformer-based detectors. In this paper, we propose Iterative Multi-scale Feature Aggregation (IMFA) – a generic paradigm that enables efficient use of multi-scale features in Transformer-based object detectors. The core idea is to exploit sparse multi-scale features from just a few crucial locations, and it is achieved with two novel designs. First, IMFA rearranges the Transformer encoder-decoder pipeline so that the encoded features can be iteratively updated based on the detection predictions. Second, IMFA sparsely samples scale-adaptive features for refined detection from just a few keypoint locations under the guidance of prior detection predictions. As a result, the sampled multi-scale features are sparse yet still highly beneficial for object detection. Extensive experiments show that the proposed IMFA boosts the performance of multiple Transformer-based object detectors significantly yet with only slight computational overhead.

1. Introduction

Detecting objects of vastly different scales has always been a major challenge in object detection [28]. Fortunately, strong evidence [11, 22, 25, 48, 69, 72] shows that object detectors can significantly benefit from multi-scale features while dealing with large scale variation. For ConvNet-based object detectors like Faster R-CNN [42] and FCOS [49], Feature Pyramid Network (FPN) [25] and its variants [12, 18, 19, 30, 48, 69, 70] have become the go-to components for exploiting multi-scale features.

Other than ConvNet-based object detectors, the recently proposed DEtection TRansformer (DETR) [4] has established a fully end-to-end object detection paradigm with promising performance. However, naively incorporating multi-scale features using FPN in these Transformer-based detectors [4, 11, 20, 29, 35, 55, 66, 72] often brings enormous and even unfeasible computation costs, primarily due to the poor efficiency of the attention mechanism in processing high-resolution features. Concretely, to handle a feature map with a spatial size of $H \times W$, ConvNet requires a computational cost of $O(HW)$, while the complexity of the attention mechanism in Transformer-based object detectors is $O(H^2W^2)$. To mitigate this issue, Deformable DETR [72] and Sparse DETR [43] replace the original global dense attention with sparse attention. SMCA-DETR [11] restricts most Transformer encoder layers to be scale-specific, with only one encoder layer to integrate multi-scale features. However, as the number of tokens increases quadratically *w.r.t.* feature map size (typically 20x∼80x compared to single-scale), these methods are still costly in computation and memory consumption, and rely on special operators like

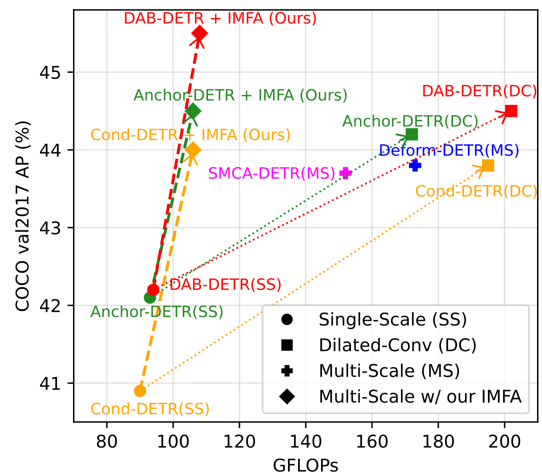

Figure 1. The proposed *Iterative Multi-scale Feature Aggregation (IMFA)* is a generic approach for efficient use of multi-scale features in Transformer-based object detectors. It boosts detection accuracy on multiple object detectors at minimal costs of additional computational overhead. Results are obtained with ResNet-50. Best viewed in color.

[*] marks corresponding author. [†] marks equal technical contribution.
Project Page: https://github.com/ZhangGongjie/IMFA.

deformable attention [72] that introduces extra complexity for deployment. To the best of our knowledge, there is yet no generic approach that can efficiently exploit multi-scale features for Transformer-based object detectors.

In this paper, we present *Iterative Multi-scale Feature Aggregation (IMFA)*, a concise and effective technique that can serve as a generic paradigm for efficient use of multi-scale features in Transformer-based object detectors. The motivation comes from two key observations: *(i)* the computation of high-resolution features is highly redundant as the background usually occupies most of the image space, thus only a small portion of high-resolution features are useful to object detection; *(ii)* unlike ConvNet, the Transformer's attention mechanism does not require grid-shaped feature maps, which offers the feasibility of aggregating multi-scale features only from some specific regions that are likely to contain objects of interest. The two observations motivate us to sparsely sample multi-scale features from just a few informative locations and then aggregate them with encoded image features in an iterative manner.

Concretely, IMFA consists of two novel designs in the Transformer-based detection pipelines. *First*, IMFA rearranges the encoder-decoder pipeline so that each encoder layer is immediately connected to its corresponding decoder layer. This design enables iterative update of encoded image features along with refined detection predictions. *Second*, IMFA sparsely samples multi-scale features from the feature pyramid generated by the backbone, with the sampling process guided by previous detection predictions. Specifically, motivated by the spatial redundancy of high-resolution features, IMFA only focuses on a few promising regions with high likelihood of object occurrence based on prior predictions. Furthermore, inspired by the significance of objects' keypoints for recognition and localization [39, 59, 66, 71], IMFA first searches several keypoints within each promising region, and then samples useful features around these keypoints at adaptively selected scales. The sampled features are finally fed to the subsequent encoder layer along with the image features encoded by the previous layer. With the two new designs, the proposed IMFA aggregates only the most crucial multi-scale features from those informative locations. Since the number of the aggregated features is small, IMFA introduces minimal computational overhead while consistently improving the detection performance of Transformer-based object detectors. It is noteworthy that IMFA is a generic paradigm for efficient use of multi-scale features: *(i)* as shown in Fig. 1, it can be easily integrated with multiple Transformer-based object detectors with consistent performance boosts; *(ii)* as discussed in Section 5.4, IMFA has the potential to boost DETR-like models on tasks beyond object detection.

To summarize, the contributions of this work are threefold.

- We propose a novel DETR-based detection pipeline, where encoded features can be iteratively updated along with refined detection predictions. This new pipeline allows to leverage intermediate predictions as guidance for robust and efficient multi-scale feature encoding.
- We propose a sparse sampling strategy for multi-scale features, which first identifies several promising regions under the guidance of prior detections, then searches several keypoints within each promising region, and finally samples their features at adaptively selected scales. We demonstrate that such sparse multi-scale features can significantly benefit object detection.
- Based on the two contributions above, we propose *Iterative Multi-scale Feature Aggregation (IMFA)* – a simple and generic paradigm that enables efficient use of multi-scale features in Transformer-based object detectors. IMFA consistently boosts detection performance on multiple object detectors, yet remains computationally efficient. This is the pioneering work that investigates a generic approach for exploiting multi-scale features efficiently in Transformer-based object detectors.

2. Related Work

Object Detection. Most modern object detectors, like Faster R-CNN [42], YOLO [40], and FCOS [49], are ConvNet-based. They have achieved promising results on various detection benchmarks [2, 7, 17, 24, 38, 44, 48, 54, 57, 61, 62]. However, these methods detect objects by defining surrogate regression and classification tasks, which rely on many hand-crafted components, such as anchors, rule-based training target assignment, and non-maximum suppression (NMS). Thus the detection pipelines of these ConvNet-based detectors are complex, hyper-parameter-intensive, and not fully end-to-end, leading to sub-optimal performance. Unlike ConvNet-based detectors, the recently proposed DETR [4] has revolutionized the paradigm for object detection using a Transformer [50] encoder-decoder architecture, eliminating the need for those hand-crafted components. Inspired by DETR [4], many Transformer-based object detectors [1, 3, 5, 8, 13, 16, 20, 23, 29, 34, 36, 45, 52, 53, 64–68, 72] are proposed and achieve state-of-the-art detection accuracy as well as fast convergence.

Multi-Scale Features for Object Detection. One major challenge in object detection is to effectively represent objects at distinct scales. This is especially crucial for detecting small objects in images. In modern ConvNet-based detectors [26, 42, 48, 49, 54, 56, 70], Feature Pyramid Network (FPN) [25] and its variants [12, 18, 30, 69, 70] have become the go-to solutions to exploit multi-scale features. However, as feature pyramids require computation on high-resolution feature maps, FPN and its variants also introduce substantial computational overhead.

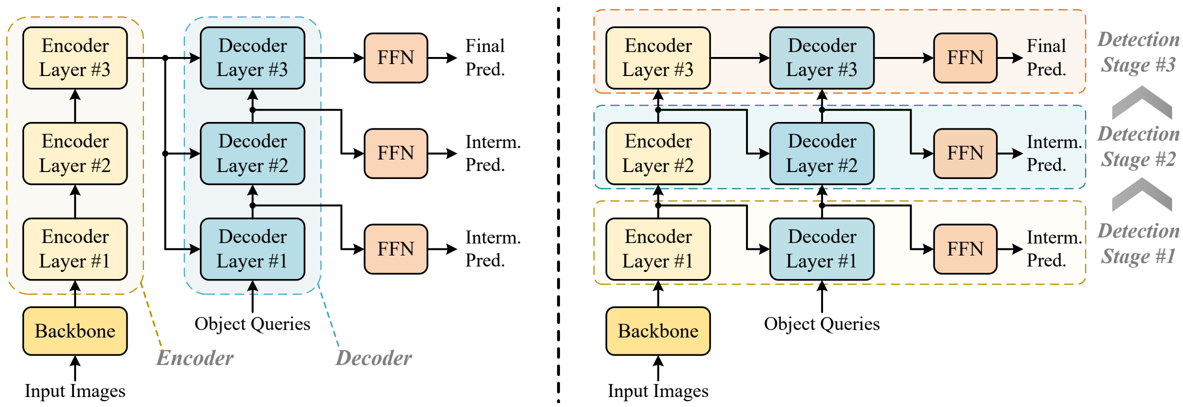

Figure 2. ***Left:*** Most existing Transformer-based object detectors employ stacked Transformer encoder layers to obtain a fixed set of encoded image features, which are fed to each Transformer decoder layer to interact with object queries. Only object queries and their corresponding detection predictions are iteratively updated. ***Right:*** IMFA rearranges the Transformer encoder-decoder pipeline into multiple stacked *detection stages*. Each *detection stage* is composed of an encoder layer, a decoder layer, and a feed-forward network (FFN), in which encoded features, object queries, and detection predictions can all be iteratively updated during the detection refinement process. Only three encoder and decoder layers are presented for concise illustration.

Multi-scale features are also helpful for Transformer-based object detectors. However, due to the inefficiency of Transformer's attention mechanism [50] to process high-resolution feature maps, it requires special modifications to reduce the computational complexity to a feasible level. Concretely, Deformable DETR [72] proposes deformable attention, which reduces the complexity via key sparsification in the attention module. SMCA-DETR [11] uses only one multi-scale attention encoder layer while restricting other layers to be scale-specific. CF-DETR [3] embeds the Transformer encoder into an FPN [25] to produce feature pyramids, and extracts multi-scale features with RoIAlign [14]. These methods enable the use of multi-scale features in Transformer-based detectors, but introduce huge computational overhead, require large-memory GPUs for training and inference, and rely on special operators like deformable attention or RoIAlign. To the best of our knowledge, there is yet no generic approach to efficiently leverage multi-scale features for Transformer-based detectors so far.

Spatial Redundancy and Sparse Features. Not all features are equally important. In most cases, only a small portion of features are crucial for recognition. With this motivation, several works [9, 10, 41, 43, 51, 52, 72] perform sparse operations over feature maps to avoid computation at less informative locations. Specifically, in object detection, AutoFocus [37] first predicts and crops regions at coarse scales, and then makes final predictions on those regions at a higher resolution. PnP-DETR [52] and Sparse DETR [43] adaptively allocate encoding operations to informative feature tokens. One similar work to our proposed IMFA is QueryDet [58], which first coarsely predicts over low-resolution features, and then sparsely exploits multi-scale features based on the coarse predictions to generate the final detection results, thus improving inference speed. However, unlike our proposed IMFA, QueryDet is designed for single-stage ConvNet-based detectors with FPN [25], and it only accelerates the inference procedure.

Our proposed IMFA is also inspired by the spatial redundancy in high-resolution features. IMFA only exploits sparse features from only a few highly informative locations to get the best of both worlds for Transformer-based detectors – high detection accuracy and low computational cost.

3. A Revisit of Transformer-Based Detection

Since our proposed method is developed on top of the recently proposed Transformer-based object detectors, we first briefly review the detection pipeline of Transformer-based object detectors [4, 29, 35, 55], taking the pioneering work DETR [4] as an example.

DETR [4] formulates object detection as a direct set prediction problem and uses a Transformer [50] encoder-decoder architecture to solve it. Given an image $\mathbf{I} \in \mathbb{R}^{H_0 \times W_0 \times 3}$, the backbone network generates its feature maps, which are further fed to the Transformer encoder to produce the encoded image features $\mathbf{F} \in \mathbb{R}^{HW \times d}$, where d denotes the feature dimension, and H_0, W_0 and H, W are the spatial sizes of the input image and its feature maps, respectively. Then, the encoded features are fed to the Transformer decoder to interact with a set of object queries representing potential objects at different spatial locations. The object queries are finally used to produce final detection predictions with a feed-forward network (FFN). The entire detection pipeline is supervised by a set-based global loss with bipartite matching.

Specifically, both the Transformer encoder and decoder are composed of multiple layers. As shown in Fig. 2 (left),

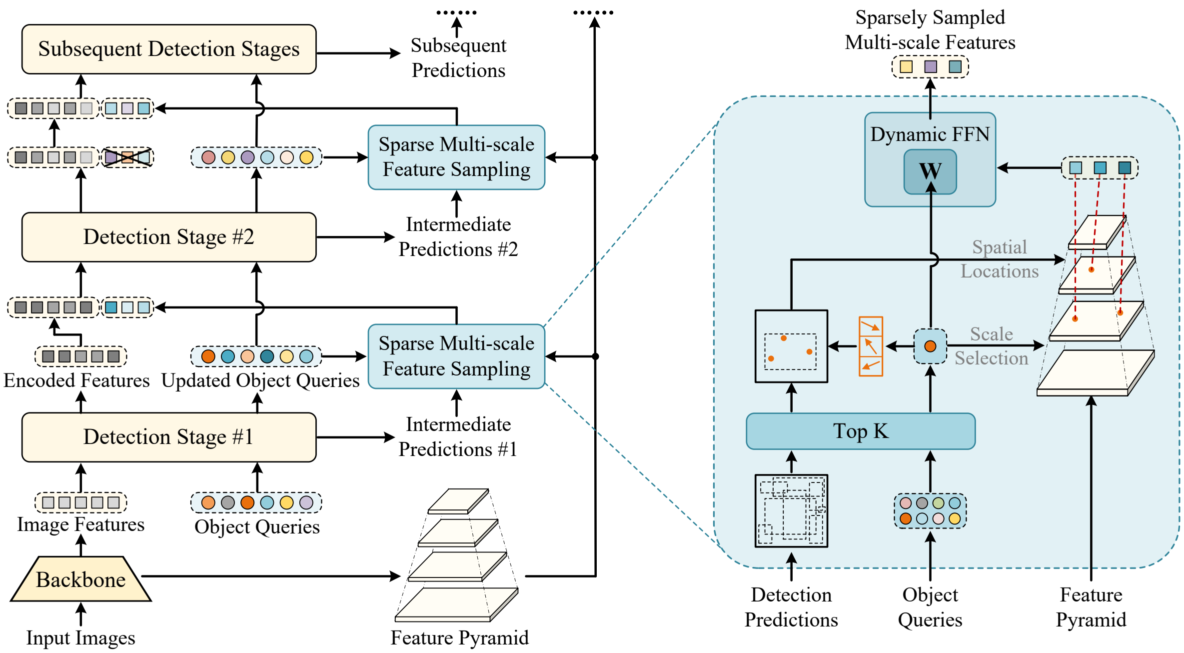

Figure 3. **The detection pipeline of *Iterative Multi-scale Feature Aggregation (IMFA)***. IMFA adopts the pipeline in Fig. 2 (right) with multiple stacked detection stages, which enables the iterative update of encoded features. On this basis, IMFA performs sparse multi-scale feature sampling under the guidance of prior detection predictions. Specifically, it only focuses on a few promising regions guided by prior detection predictions, then searches for several keypoints within each promising region, and finally samples features around these keypoints at adaptively selected scales. IMFA also adopts a Dynamic FFN to enhance the representation capacity of sparsely sampled multi-scale features by incorporating semantics from their corresponding object queries. The sampled features are fed into the subsequent detection stages along with encoded features for refined detection. Only the first two detection stages are presented for concise illustration.

existing methods [4,11,29,35,55,72] usually process the input image features with a stack of encoder layers and obtain a fixed set of encoded features, which are further fed to the Transformer decoder layers to update the detection results iteratively. Differently, as illustrated in Fig. 2 (right), one major difference introduced by IMFA is that it rearranges the encoder-decoder pipeline into multiple stacked detection stages, so that encoded features can also be iteratively updated along with refined detection predictions. This design modification lays the foundation for efficient use of multi-scale features guided by prior detection results, which is to be detailed in the next section.

4. Iterative Multi-Scale Feature Aggregation

4.1. Overview

Iterative Multi-scale Feature Aggregation (IMFA) is a generic paradigm for efficient use of multi-scale features in Transformer-based object detectors, such as DETR [4]. Fig. 3 illustrates the detection pipeline of the proposed IMFA. For computational efficiency, IMFA exploits multi-scale features with dual-sparsity: *(i)* it samples multi-scale features from just a few promising regions with high likelihood of object occurrence as guided by prior detection predictions; *(ii)* for each promising region, it only samples features from several keypoints with the most informative features at adaptively selected scales. The dual-sparsity is achieved with two novel designs, which are to be described in detail in the following subsections.

4.2. Iterative Update of Encoded Features

The iterative update of encoded image features is the basis for IMFA to exploit multi-scale features efficiently. As introduced in Section 3, most existing Transformer-based detectors use fixed encoded image features to make predictions. In order to guide the multi-scale sampling process with prior detections, IMFA rearranges the Transformer encoder-decoder pipeline, as shown in Fig. 2 (right).

Specifically, instead of using stacked encoder layers to produce a fixed set of feature tokens at one go, IMFA rearranges the detection pipeline into several stacked *detection stages*. Each *detection stage* consists of an encoder layer, a decoder layer, and an FFN. This design lays the foundation for incorporating sparse multi-scale features dynamically under the guidance of prior detection predictions, which is detailed in Section 4.3. It is noteworthy that, according to the experiments in Section 5.3, this design alone (shown in Fig. 2 (right), without incorporating multi-scale features) brings no performance gain over the baseline model.

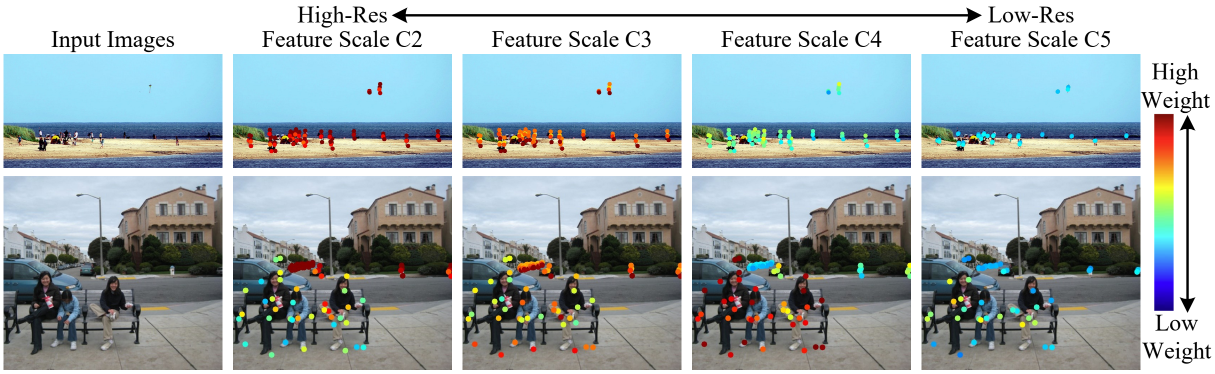

Figure 4. **Visualization of IMFA's sampling locations and their adaptively selected feature scales.** The searched sampling points mostly fall around the objects of interest, many of which are highly representative points with rich semantics, such as objects' extremities. Besides, IMFA adaptively selects appropriate feature scales for each sampling point, generating sparse yet informative scale-adaptive features for refined detection predictions. Best viewed in color. More visualizations are provided in technical appendix.

4.3. Sparse Feature Sampling and Aggregation

Naively incorporating multi-scale features into the encoder leads to prohibitive computational complexity, as the number of feature tokens from all scales is too large to be processed by the attention mechanism. This motivates us to exploit only the most informative multi-scale features.

On the basis of Section 4.2, IMFA further performs sparse multi-scale feature sampling using prior detection predictions as guidance, as illustrated in Fig. 3. Specifically, IMFA first identifies a few promising regions with high likelihood of object occurrence. Then, it searches for several representative and informative keypoints within each promising region and samples their features at adaptively selected scales. Finally, the sampled features are fed to the subsequent encoder layers to aggregate with single-scale image features to produce refined detection predictions.

Identifying Promising Regions Based on Prior Predictions. In most cases, objects are sparsely distributed across images [27, 37, 58], which motivates us to exploit only the multi-scale features related to these objects. An intuitive solution is to guide the sampling process with the high-confidence detection predictions from the previous detection stage. Concretely, as shown in Fig. 3, for each detection stage except the first stage, we select K predictions with the highest classification confidence scores from the previous detection stage as the promising regions. Here, $K = N \times r$, with N denoting the number of object queries and r denoting IMFA's sampling ratio. Formally, we denote the selected box predictions and their corresponding object queries as $\{(\mathbf{B}_1, \mathbf{Q}_1), ..., (\mathbf{B}_K, \mathbf{Q}_K)\}$. The multi-scale features are then sampled within these promising regions, which is to be introduced in detail later. Since Transformer-based object detectors [4, 11, 29, 35, 55] already employ a sparse set (typically 100∼300) of object queries to represent different objects, the promising regions sampled by IMFA remain sparse for efficient computation.

Sampling Scale-Adaptive Features from Representative Keypoints. IMFA directly samples multi-scale features from the feature pyramid that is generated from the backbone (C2-C5 from ResNet in our experiments). However, even the sparsely sampled promising regions still contain a substantial amount of feature tokens at high-resolution feature scales. To further sparsify the sampled multi-scale features, IMFA searches a small number of representative keypoints within each promising region and samples their corresponding features at adaptively selected scales.

As illustrated in Fig. 3, for each promising region, IMFA first uses its object query to predict M keypoint locations within the region, which can be formulated as:

$$\{P_{ij}\}_{j=1}^{M} = \text{MLP}(\mathbf{Q}_i) \quad \text{for } i = 1, 2, ..., K \; , \tag{1}$$

where i and j index the queries and keypoints, respectively, and each keypoint $P_{ij} = (x_{ij}, y_{ij})$ lies within its corresponding box prediction $\mathbf{B}_i$. Then, IMFA samples each keypoint's features from the feature pyramid at all scales via bilinear interpolation, obtaining a set of features $\{\mathbf{F}_{ij}^{s}\}_{s=1}^{S}$, where S is the number of feature scales. Finally, to emphasize the distinct significance of different feature scales for each keypoint, we propose to perform adaptive scale selection by predicting scale-specific weights for each keypoint and obtaining scale-adaptive features through weighted summation:

$$\mathbf{F}_{ij} = \textstyle\sum_s \alpha_{ij}^{s} \mathbf{F}_{ij}^{s} \qquad \{\alpha_{ij}^{s}\}_{s=1}^{S} = \text{Softmax}(\gamma_j(\mathbf{Q}_i)), \tag{2}$$

where the scale-selection weights α are generated by a linear projection γ_j followed by a Softmax function, so that $\sum_s \alpha_{ij}^{s} = 1$. In this way, IMFA only samples the most crucial and informative features, producing a set of sparse yet still highly informative multi-scale features for each promising region. Additionally, to further strengthen the representation capacity of the sampled multi-scale features, we feed the sampled features into a Dynamic Feed-

Method	High-Res Feat	#Epochs	#Params	FLOPs	FPS	GPU Mem	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
DETR-R50 [4] ‡		50	41M	86G	24.6	2.1 GB	34.9	55.5	36.0	14.4	37.2	54.5
DETR-R50-DC5 [4] ‡	✓	50	41M	187G	9.2	5.8 GB	36.7	57.6	38.2	15.4	39.8	**56.3**
DETR-R50 [4] + IMFA (Ours) ‡		50	52M	105G	20.0	2.5 GB	**39.2**	**58.8**	**41.6**	**20.3**	**42.2**	55.4
Conditional-DETR-R50 [35]		50	44M	90G	22.2	2.1 GB	40.9	61.8	43.3	20.8	44.6	59.2
Conditional-DETR-R50-DC5 [35]	✓	50	44M	195G	8.9	5.8 GB	43.8	**64.4**	46.7	24.0	**47.6**	**60.7**
Conditional-DETR-R50 [35] + IMFA (Ours)		50	53M	106G	19.0	2.5 GB	**44.0**	64.2	**47.5**	**25.7**	46.8	59.8
Anchor-DETR-R50 [55]		50	37M	93G	22.3	2.1 GB	42.1	63.1	44.9	22.3	46.2	60.0
Anchor-DETR-R50-DC5 [55]	✓	50	37M	172G	14.3	3.6 GB	44.2	**64.7**	47.5	24.7	**48.2**	**60.6**
Anchor-DETR-R50 [55] + IMFA (Ours)		50	46M	106G	17.5	2.4 GB	**44.5**	63.9	**47.7**	**26.4**	47.7	59.9
DAB-DETR-R50 [29]		50	44M	94G	21.4	2.1 GB	42.2	63.1	44.7	21.5	45.7	60.3
DAB-DETR-R50-DC5 [29]	✓	50	44M	202G	8.8	6.0 GB	44.5	**65.1**	47.7	25.3	48.2	**62.3**
DAB-DETR-R50 [29] + IMFA (Ours)		50	53M	108G	18.6	2.5 GB	**45.5**	65.0	**49.3**	**27.3**	**48.3**	61.6

Table 1. Compatibility with different Transformer-based object detectors. IMFA boosts the performance of existing detectors at slight computational costs. 'High-Res Feat' denotes the use of high-resolution features with R50-DC5. ‡ denotes DETR with 300 object queries and focal loss. Results are reported on COCO val 2017.

Forward Network (Dynamic FFN) to incorporate the semantics from their corresponding object queries via dynamic weighting [46], where FFN's weights are dynamically generated by object queries. It can be formulated as:

$$\mathbf{F}'_{ij} = \mathrm{MLP}_{\mathbf{W}_i}(\mathbf{F}_{ij}) \qquad \mathbf{W}_i = \psi(\mathbf{Q}_i). \tag{3}$$

Here, for each object query $\mathbf{Q}_i$, the dynamic weight $\mathbf{W}_i$ is obtained by a linear projection ψ of $\mathbf{Q}_i$. Then, $\mathbf{W}_i$ is applied to the scale-adaptive features $\mathbf{F}_{ij}$ to generate the final sampled features $\mathbf{F}'_{ij}$ with enhanced semantics. These sampled features, along with their positional embeddings obtained based on their keypoint locations, are further fed to the subsequent detection stage for aggregation.

Iterative Aggregation of Multi-Scale Features. To leverage the sampled multi-scale features for refined object detection, the sampled features and the encoded image features are fed into the subsequent encoder layer for aggregation using the attention mechanism. This is analogous to the top-down path created by FPN [25] for enhancing the semantics of low-level features. To avoid continuous growth of feature tokens and maintain efficiency, each detection stage does not inherit the multi-scale features that are generated from the previous stage, as shown in Fig. 3.

4.4. Visualization and Analysis

Fig. 4 visualizes IMFA's sampling locations and their feature scales. It can be observed that the sampling locations mostly fall around the target objects, and typically at representative locations, such as object extremities. This proves the effectiveness of IMFA in searching sparse yet highly informative locations in the feature sampling process. Besides, it is noteworthy that IMFA tends to focus on higher-resolution features for small objects and lower-resolution features for large objects, which is intuitive as the detection of small objects relies more on finer details.

5. Experiments

5.1. Experiment Setup

Dataset and Evaluation Metrics. We perform experiments on the COCO 2017 dataset [27]. We use ∼117k images in `train2017` for training and 5k images in `val2017` for evaluation. We adopt COCO's standard evaluation metrics for performance evaluation.

Implementation Details. As the proposed IMFA defines a generic paradigm, we mainly conduct experiments with DAB-DETR [29] – a state-of-the-art Transformer-based object detector with open-sourced implementation. We also integrate IMFA with DETR [4], Conditional DETR [35], and Anchor DETR [55], to demonstrate its generality.

A crucial implementation detail involves incorporating skip connections for encoded features between Transformer encoder layers, as motivated by [63] and [65, 66] to facilitate feature semantic alignment.

For IMFA-related hyper-parameters, we set the sampling ratio r at 20% and the keypoint number M at 8 by default. Other model-related setups align with their corresponding baselines [4, 29, 35, 55]. We use ImageNet-pretrained [6] ResNet [15] as backbone networks, and conduct training with AdamW optimizer [33]. The total batch size is set to 16 for training. The initial learning rate is 1×10^{-5} for the backbone networks and 1×10^{-4} for the Transformer architectures, along with a weight decay of 1×10^{-4}. Models are trained for 50 epochs, with the learning rate decayed at the 40^{th} epoch by 0.1. The same data augmentation scheme used in [4, 29, 35, 55] is adopted.

5.2. Experiment Results

Compatibility with Transformer-Based Detectors. We first evaluate the generality of IMFA by integrating it with multiple Transformer-based object detectors. As discussed in Section 1, these methods resort to higher-resolution backbones (denoted with 'High-Res Feat') as an alternative, as it

Method	MS	SMS	DC	#Epochs	#Params	FLOPs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster-RCNN-FPN-R50 [25, 42]	✓			108	42M	180G	42.0	62.1	45.5	26.6	45.5	53.4
TSP-FCOS-FPN-R50 [47]	✓			36	52M	189G	43.1	62.3	47.0	26.6	46.8	55.9
TSP-RCNN-FPN-R50 [47]	✓			36	64M	188G	43.8	63.3	48.3	**28.6**	46.9	55.7
Sparse-RCNN-FPN-R50 [46]	✓			36	106M	166G	45.0	64.1	48.9	28.0	47.6	59.5
DETR-R50 [4]			✓	500	41M	187G	43.3	63.1	45.9	22.5	47.3	61.1
Deformable-DETR-R50 [72]	✓			50	40M	173G	43.8	62.6	47.7	26.4	47.1	58.0
Deformable-DETR-R50 [72] + Iter	✓			50	41M	173G	45.4	64.7	49.0	26.8	48.3	61.7
Efficient-DETR-R50 [60]	✓			36	32M	159G	44.2	62.2	48.0	28.4	47.5	56.6
Conditional-DETR-R50 [35]			✓	50	44M	195G	43.8	64.4	46.7	24.0	47.6	60.7
SMCA-DETR-R50 [11]	✓			50	40M	152G	43.7	63.6	47.2	24.2	47.0	60.4
YOLOS-DeiT-S [8]				150	28M	172G	37.6	57.6	39.2	15.9	40.2	57.3
Anchor-DETR-R50 [55]			✓	50	37M	172G	44.2	64.7	47.5	24.7	48.2	60.6
DAB-DETR-R50 [29]			✓	50	44M	202G	44.5	65.1	47.7	25.3	48.2	62.3
SAM-DETR-R50 [66]			✓	50	58M	210G	43.3	64.4	46.2	25.1	46.9	61.0
SAM-DETR-R50 [66] w/ SMCA [11]			✓	50	58M	210G	45.0	**65.4**	47.9	26.2	**49.0**	**63.3**
DAB-DETR-R50 [29] + IMFA (Ours)		✓		50	53M	**108G**	**45.5**	65.0	**49.3**	27.3	48.3	61.6

Table 2. Comparison with state-of-the-art object detectors on COCO val 2017. Our proposed method achieves comparable performance with the state-of-the-art methods, but with significantly lower computation. 'MS' denotes the use of multi-scale features. 'SMS' denotes the use of sparse multi-scale features with our proposed IMFA. 'DC' denotes the use of high-resolution features with R50-DC5.

Method	#Params	FLOPs	FPS	AP
DETR-SwinB [4]	105M	303G	9.8	40.7
DETR-SwinB [4] + IMFA (Ours)	115M	318G	9.3	**46.2**
DAB-DETR-ConvNextB [29]	108M	287G	9.4	47.4
DAB-DETR-ConvNextB [29] + IMFA (Ours)	117M	301G	8.7	**50.0**

Table 3. Results under stronger backbones. Results are obtained on COCO val 2017.

Iter. Enc.	SFSA	#Params	FLOPs	AP	AP_S	AP_M	AP_L
		44M	94G	42.2	21.5	45.7	60.3
✓		44M	94G	41.9	21.8	45.2	61.1
✓	✓	53M	108G	**45.5**	**27.3**	**48.3**	**61.6**

Table 4. Ablation studies on IMFA's two major design choices. 'Iter. Enc.' denotes iterative update of encoded features as illustrated in Fig. 2 (right). 'SFSA' denotes sparse feature sampling and aggregation as illustrated in Fig. 3.

Rep. Kp.	Ada. Scale	Dy. FFN	#Params	FLOPs	AP	AP_S	AP_M	AP_L
			44M	94G	41.9	21.8	45.2	61.1
	✓		45M	105G	42.1	22.0	45.4	61.0
	✓	✓	53M	108G	42.3	22.2	46.0	60.9
✓		✓	53M	108G	44.7	26.4	47.6	61.5
✓	✓		45M	105G	44.2	26.3	47.2	60.8
✓	✓	✓	53M	108G	**45.5**	**27.3**	**48.3**	**61.6**

Table 5. Ablation studies on the design choices within sparse multi-scale feature sampling and aggregation. 'Rep. Kp.' denotes searching representative keypoints. 'Ada. Scale' denotes adaptive scale selection. 'Dy. FFN' denotes Dynamic FFN.

is computationally prohibitive for them to directly process multi-scale features. As shown in Table 1, using higher-resolution features improves the detection performance but adds a substantial computational cost (+ ∼100 GFLOPs and - 8∼15 FPS) as well as GPU memory consumption. On the other hand, the proposed IMFA consistently improves the detection performance by large margins across all metrics, especially on small objects (AP_S), yet only introduces a slight computational overhead (+ ∼15 GFLOPs and - ∼3 FPS). The experimental results demonstrate IMFA's effectiveness and wide applicability.

Comparison with State-of-the-Art Detectors. We integrate IMFA with DAB-DETR [29] to benchmark with other state-of-the-art single-stage Transformer-based detectors that utilize high-resolution or multi-scale features. We also include some popular two-stage detectors [42, 47, 60] for a comprehensive comparison. As shown in Table 2, our method can achieve comparable performance with the state-of-the-art methods, but with significantly less computational cost.

Results with Stronger Backbones. As shown in Table 3, when using stronger backbones [31, 32], IMFA still consistently improves detection performance at marginal costs.

5.3. Ablation Study

We conduct ablation studies with the strong baseline DAB-DETR-R50 [15, 29] to validate the effectiveness of our designs. Results are obtained on COCO val 2017.

Effect of IMFA's Design Choices. IMFA introduces two novel designs: *i)* iterative encoding described in Section 4.2 and Fig. 2 (right), and *ii)* sparse multi-scale feature sampling and aggregation described in Section 4.3 and Fig. 3. As shown in Table 4, the iterative encoding alone even slightly degrades the baseline's performance. However, with IMFA's sparsely sampled multi-scale features, our method significantly improves the detection performance of objects at all scales, especially at smaller scales. This proves that the multi-scale features sampled by IMFA are sparse yet highly effective for object detection.

We also study the three crucial components within the sparse feature sampling and aggregation process in Table 5. Without identifying representative keypoints (random spa-

r	#Params	FLOPs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
10%	53M	103G	44.2	64.0	47.5	25.9	47.3	60.6
15%	53M	105G	44.8	64.2	48.2	26.5	47.7	60.1
20%	53M	108G	**45.5**	65.0	**49.3**	27.3	**48.3**	**61.6**
25%	53M	111G	45.3	**65.1**	49.0	27.9	47.9	61.1
30%	53M	114G	45.1	64.5	48.9	**28.4**	48.2	60.2

Table 6. Ablation study on the sampling ratio r of prior detection predictions. Results are obtained on COCO val 2017.

M	#Params	FLOPs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
1	53M	101G	43.9	64.3	47.5	25.1	46.9	60.8
2	53M	102G	45.0	64.7	48.9	26.0	48.3	60.4
4	53M	104G	45.3	**65.0**	48.7	**27.3**	48.1	60.9
8	53M	108G	**45.5**	**65.0**	**49.3**	**27.3**	48.3	**61.6**
16	53M	117G	45.3	64.7	49.0	26.6	**48.5**	61.5

Table 7. Ablation study on the keypoint number M within each promising region. Results are obtained on COCO val 2017.

tial sampling is used instead), the performance barely improves, which verifies our claim that only a very small set of multi-scale features are beneficial. The results also validate that IMFA can search keypoints with important semantics information. Without adaptive scale selection (averaged scale selection is used instead), the performance drops, indicating that our design enables the focus of appropriate scales for each object. Without Dynamic FFN, the performance also drops, which proves that Dynamic FFN successfully fuses important semantics information from the corresponding object queries and benefits the final prediction.

Effect of IMFA's Hyper-Parameters. IMFA introduces two hyper-parameters: the sampling ratio of prior detection predictions and object queries (r) as well as the keypoint number in each promising region (M). We conduct sensitivity analysis on each of them.

Table 6 shows the effect of different r values when M is fixed at 8. As r increases from 10% to 30%, the average precision (AP) first increases then decreases, while the computational cost keeps growing. An interesting trend is that the detection performance of small objects (AP_S) goes up with increasing r consistently. We conjecture that small objects rely more on the fine details in high-resolution features, so that they can benefit from increased number of promising regions used for multi-scale feature sampling. However, the overall performance drops when r is too large, which we conjecture is due to the increased difficulty in searching relevant features with overwhelming featuare tokens involved. Based on the experimental results, we set the default value for r as 20% in our system.

To study the effect of the number of keypoints M, we conduct experiments by fixing r at 20% and report the results in Table 7. We can see a similar trend that the performance improves as M increases but then drops when M becomes too large. Therefore, we set M as 8 by default.

Method	Input Size	FLOPs	FPS	AP^{kp}
PRTR-R50 [21]	384x288	11.0 G	360	68.2
PRTR-R50 [21]	512x384	18.8 G	218	71.0
PRTR-R50 [21] + IMFA (Ours)	384x288	13.4 G	293	**72.7**
PRTR-R101 [21]	384x288	19.1 G	243	70.1
PRTR-R101 [21]	512x384	33.4 G	144	72.0
PRTR-R101 [21] + IMFA (Ours)	384x288	21.5 G	216	**73.7**

Table 8. Human pose estimation performance on COCO val 2017. IMFA greatly boosts performance at marginal costs, even surpassing the baseline methods with high-resolution input images.

5.4. Extension to Human Pose Estimation

We further apply the proposed IMFA to human pose estimation to verify its generality across different tasks. Concretely, we evaluate the performance on the COCO 2017 human pose estimation benchmark [27]. We adopt PRTR (two-stage variant) [21], a DETR-like human pose estimation method with open-sourced implementation, as our baseline. Please refer to the technical appendix for its full implementation details.

As shown in Table 8, on the task of human pose estimation, IMFA still clearly outperforms its baseline methods at the same input size with only slight extra computation. IMFA even surpasses its higher-resolution baselines at significantly reduced computational costs. The results indicate IMFA's potential of boosting Transformer-based models on various vision tasks beyond object detection itself.

6. Conclusion

Multi-scale features are beneficial to object detection, but often come with large computational costs. This paper presents *Iterative Multi-scale Feature Aggregation (IMFA)* as the pioneering generic paradigm for efficient use of multi-scale features in Transformer-based object detectors. It gets the best of both worlds – high accuracy and low computational cost. IMFA identifies and extracts multi-scale features from the most promising and informative locations only and greatly improves detection accuracy on multiple object detectors at marginal additional costs. We expect IMFA will inspire more comprehensive research and applications on Transformer-based object detection.

Limitations. Although IMFA is compatible with many Transformer-based object detectors, it cannot be directly applied to Deformable DETR [72] and its extensions [43, 60]. This is due to undefined deformable operations on non-grid feature maps, which require extensive engineering efforts.

Acknowledgement:
This study is supported under the RIE 2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s).

References

[1] Amir Bar, Xin Wang, Vadim Kantorov, Colorado J Reed, Roei Herzig, Gal Chechik, Anna Rohrbach, Trevor Darrell, and Amir Globerson. DETReg: Unsupervised pretraining with region priors for object detection. In *CVPR*, 2022. 2

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In *CVPR*, 2018. 2

[3] Xipeng Cao, Peng Yuan, Bailan Feng, and Kun Niu. CF-DETR: Coarse-to-fine transformers for end-to-end object detection. In *AAAI*, 2022. 2, 3

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with Transformers. In *ECCV*, 2020. 1, 2, 3, 4, 5, 6, 7

[5] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. UP-DETR: Unsupervised pre-training for object detection with transformers. In *CVPR*, 2021. 2

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[7] Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai. Few-shot object detection with attention-RPN and multi-relation detector. In *CVPR*, 2020. 2

[8] Yuxin Fang, Bencheng Liao, Xinggang Wang, Jiemin Fang, Jiyang Qi, Rui Wu, Jianwei Niu, and Wenyu Liu. You only look at one sequence: Rethinking transformer in vision through object detection. In *NeurIPS*, 2021. 2, 7

[9] Michael Figurnov, Maxwell D Collins, Yukun Zhu, Li Zhang, Jonathan Huang, Dmitry Vetrov, and Ruslan Salakhutdinov. Spatially adaptive computation time for residual networks. In *CVPR*, 2017. 3

[10] Mikhail Figurnov, Aizhan Ibraimova, Dmitry P Vetrov, and Pushmeet Kohli. PerforatedCNNs: Acceleration through elimination of redundant convolutions. In *NeurIPS*, 2016. 3

[11] Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fast convergence of DETR with spatially modulated co-attention. In *ICCV*, 2021. 1, 3, 4, 5, 7

[12] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. NAS-FPN: Learning scalable feature pyramid architecture for object detection. In *CVPR*, 2019. 1, 2

[13] Akshita Gupta, Sanath Narayan, KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. OW-DETR: Open-world detection transformer. In *CVPR*, 2022. 2

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 3

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6, 7

[16] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. MDETR–modulated detection for end-to-end multi-modal understanding. In *ICCV*, 2021. 2

[17] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *ICCV*, 2019. 2

[18] Seung-Wook Kim, Hyong-Keun Kook, Jee-Young Sun, Mun-Cheon Kang, and Sung-Jea Ko. Parallel feature pyramid network for object detection. In *ECCV*, 2018. 1, 2

[19] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *CVPR*, 2019. 1

[20] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. DN-DETR: Accelerate DETR training by introducing query denoising. In *CVPR*, 2022. 1, 2

[21] Ke Li, Shijie Wang, Xiang Zhang, Yifan Xu, Weijian Xu, and Zhuowen Tu. Pose recognition with cascade transformers. In *CVPR*, 2021. 8

[22] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. In *ICCV*, 2019. 1

[23] Bencheng Liao, Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Wenyu Liu, and Chang Huang. MapTR: Structured modeling and learning for online vectorized HD map construction. In *ICLR*, 2023. 2

[24] Minghui Liao, Pengyuan Lyu, Minghang He, Cong Yao, Wenhao Wu, and Xiang Bai. Mask TextSpotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(2):532–548, 2021. 2

[25] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 1, 2, 3, 6, 7

[26] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 2

[27] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 5, 6, 8

[28] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *International Journal of Computer Vision*, 128:261–318, 2020. 1

[29] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. DAB-DETR: Dynamic anchor boxes are better queries for DETR. In *ICLR*, 2022. 1, 2, 3, 4, 5, 6, 7

[30] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *CVPR*, 2018. 1, 2

[31] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical vision Transformer using shifted windows. In *ICCV*, 2021. 7

[32] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *CVPR*, 2022. 7

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

[34] Zhipeng Luo, Gongjie Zhang, Changqing Zhou, Tianrui Liu, Shijian Lu, and Liang Pan. TransPillars: Coarse-to-fine aggregation for multi-frame 3D object detection. In *WACV*, 2023. 2

[35] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional DETR for fast training convergence. In *ICCV*, 2021. 1, 3, 4, 5, 6, 7

[36] Ishan Misra, Rohit Girdhar, and Armand Joulin. An End-to-End Transformer Model for 3D Object Detection. In *ICCV*, 2021. 2

[37] Mahyar Najibi, Bharat Singh, and Larry S Davis. AutoFocus: Efficient multi-scale inference. In *ICCV*, 2019. 3, 5

[38] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra R-CNN: Towards balanced learning for object detection. In *CVPR*, 2019. 2

[39] Han Qiu, Yuchen Ma, Zeming Li, Songtao Liu, and Jian Sun. BorderDet: Border feature for dense object detection. In *ECCV*, 2020. 2

[40] Joseph Redmon and Ali Farhadi. YOLO 9000: Better, faster, stronger. In *CVPR*, 2017. 2

[41] Mengye Ren, Andrei Pokrovsky, Bin Yang, and Raquel Urtasun. SBNet: Sparse blocks network for fast inference. In *CVPR*, 2018. 3

[42] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 1, 2, 7

[43] Byungseok Roh, JaeWoong Shin, Wuhyun Shin, and Saehoon Kim. Sparse DETR: Efficient end-to-end object detection with learnable sparsity. In *ICLR*, 2022. 1, 3, 8

[44] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. In *CVPR*, 2019. 2

[45] Hwanjun Song, Deqing Sun, Sanghyuk Chun, Varun Jampani, Dongyoon Han, Byeongho Heo, Wonjae Kim, and Ming-Hsuan Yang. ViDT: An efficient and effective fully transformer-based object detector. In *ICLR*, 2022. 2

[46] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, and Ping Luo. Sparse R-CNN: End-to-end object detection with learnable proposals. In *CVPR*, 2021. 6, 7

[47] Zhiqing Sun, Shengcao Cao, Yiming Yang, and Kris M. Kitani. Rethinking Transformer-based set prediction for object detection. In *ICCV*, 2021. 7

[48] Mingxing Tan, Ruoming Pang, and Quoc V Le. EfficientDet: Scalable and efficient object detection. In *CVPR*, 2020. 1, 2

[49] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully convolutional one-stage object detection. In *ICCV*, 2019. 1, 2

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, L. Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2, 3

[51] Thomas Verelst and Tinne Tuytelaars. Dynamic convolutions: Exploiting spatial sparsity for faster inference. In *CVPR*, 2020. 3

[52] Tao Wang, Li Yuan, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. PnP-DETR: Towards efficient visual analysis with Transformers. In *ICCV*, 2021. 2, 3

[53] Wen Wang, Yang Cao, Jing Zhang, and Dacheng Tao. FP-DETR: Detection transformer advanced by fully pretraining. In *ICLR*, 2022. 2

[54] Xin Wang, Thomas E. Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. In *ICML*, 2020. 2

[55] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor DETR: Query design for Transformer-based detector. In *AAAI*, 2022. 1, 3, 4, 5, 6, 7

[56] Jiaxi Wu, Songtao Liu, Di Huang, and Yunhong Wang. Multi-scale positive sample refinement for few-shot object detection. In *ECCV*, 2020. 2

[57] Yang Xiao and Renaud Marlet. Few-shot object detection and viewpoint estimation for objects in the wild. In *ECCV*, 2020. 2

[58] Chenhongyi Yang, Zehao Huang, and Naiyan Wang. QueryDet: Cascaded sparse query for accelerating high-resolution small object detection. In *CVPR*, 2022. 3, 5

[59] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. RepPoints: Point set representation for object detection. In *ICCV*, 2019. 2

[60] Zhuyu Yao, Jiangbo Ai, Boxun Li, and Chi Zhang. Efficient DETR: improving end-to-end object detector with dense prior. *arXiv preprint arXiv:2104.01318*, 2021. 7, 8

[61] Gongjie Zhang, Kaiwen Cui, Rongliang Wu, Shijian Lu, and Yonghong Tian. PNPDet: Efficient few-shot detection without forgetting via plug-and-play sub-networks. In *WACV*, 2021. 2

[62] Gongjie Zhang, Shijian Lu, and Wei Zhang. CAD-Net: A context-aware detection network for objects in remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 57(12):10015–10024, 2019. 2

[63] Gongjie Zhang, Zhipeng Luo, Kaiwen Cui, and Shijian Lu. Meta-DETR: Few-shot object detection via unified image-level meta-learning. *arXiv preprint arXiv:2103.11731v1*, 2021. 6

[64] Gongjie Zhang, Zhipeng Luo, Kaiwen Cui, Shijian Lu, and Eric P. Xing. Meta-DETR: Image-level few-shot detection with inter-class correlation exploitation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2

[65] Gongjie Zhang, Zhipeng Luo, Jiaxing Huang, Shijian Lu, and Eric P Xing. Semantic-aligned matching for enhanced DETR convergence and multi-scale feature fusion. *arXiv preprint arXiv:2207.14172*, 2022. 2, 6

[66] Gongjie Zhang, Zhipeng Luo, Yingchen Yu, Kaiwen Cui, and Shijian Lu. Accelerating DETR convergence via semantic-aligned matching. In *CVPR*, 2022. 1, 2, 6, 7

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

[68] Jingyi Zhang, Jiaxing Huang, Zhipeng Luo, Gongjie Zhang, Xiaoqin Zhang, and Shijian Lu. DA-DETR: Domain Adaptive Detection Transformer with information fusion. In *CVPR*, 2023. 2

[69] Gangming Zhao, Weifeng Ge, and Yizhou Yu. GraphFPN: Graph feature pyramid network for object detection. In *ICCV*, 2021. 1, 2

[70] Qijie Zhao, Tao Sheng, Yongtao Wang, Zhi Tang, Ying Chen, Ling Cai, and Haibin Ling. M2Det: A single-shot object

detector based on multi-level feature pyramid network. In *AAAI*, 2019. 1, 2

[71] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krähenbühl. Bottom-up object detection by grouping extreme and center points. In *CVPR*, 2019. 2

[72] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *ICLR*, 2021. 1, 2, 3, 4, 7, 8