Rethinking Object Detection Framework through the Lens of Proposal Refinement

JINGJING ZHAO

B.Eng.

Lead Supervisor: Prof. Chang Xu
Supervisor: Dr. Siqi Ma

A thesis submitted in fulfilment of
the requirements for the degree of
Master of Philosophy

School of Computer Science
Faculty of Engineering
The University of Sydney
Australia

21 April 2025

Statement of Originality

This is to certify that to the best of my knowledge, the content of this thesis is my own work. This thesis has not been submitted for any degree or other purposes.

I certify that the intellectual content of this thesis is the product of my own work and that all the assistance received in preparing this thesis and sources have been acknowledged.

Student: Jingjing Zhao

Signature: **Date**:

Authorship Attribution Statement

This thesis was conducted at the University of Sydney, under the supervision of Prof. Chang Xu, during 2024. The main results presented in this dissertation were first introduced in the following publication:

- **Zhao, J**., Wei, F., & Xu, C. (2024). Hybrid Proposal Refiner: Revisiting DETR Series from the Faster R-CNN Perspective. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 17416-17426). Presented in Chatper 3. I designed the main techniques, implemented the system, designed and conducted the evaluation, and wrote the draft manuscript.

Student: Jingjing Zhao

Signature: **Date**:

As the supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Supervisor: Prof. Chang Xu

Signature: **Date**:

Abstract

With the transformative impact of the Transformer architecture, DETR pioneered the application of the encoder-decoder framework to object detection. Subsequent research, such as Deformable DETR, has aimed to enhance DETR while maintaining the encoder-decoder design. In this thesis, we revisit the DETR series through the lens of Faster R-CNN. We discover that DETR aligns with the underlying principles of Faster R-CNN's RPN-refiner design but gains advantages in end-to-end detection through the incorporation of Hungarian matching.

We systematically adapt Faster R-CNN towards Deformable DETR by integrating or repurposing each component of Deformable DETR within the Faster R-CNN framework. Our thorough analysis demonstrates that Deformable DETR's improved performance over Faster R-CNN is primarily attributable to the adoption of advanced modules, such as a superior proposal refiner that utilizes deformable attention mechanisms instead of traditional techniques like RoI Align. By viewing DETR through the RPN-refiner paradigm, we explore various proposal refinement techniques, including deformable attention, cross attention, and dynamic convolution. Each of these proposal refiners offers unique strengths in accurately refining object proposals by dynamically adjusting the focus and processing regions of interest. Our empirical studies indicate that these proposal refiners complement each other effectively, leading us to synergistically combine them into a Hybrid Proposal Refiner (HPR), which leverages the strengths of each refinement technique to enhance overall detection performance.

Our HPR is designed to be highly versatile and can be seamlessly incorporated into various DETR-based detectors, enhancing their detection capabilities. For instance, by integrating HPR into a strong DETR detector, we achieve an impressive Average Precision (AP) of 54.9 on the COCO benchmark, utilizing a ResNet-50 backbone and a 36-epoch training schedule. This significant improvement underscores the effectiveness of HPR in refining proposals and boosting detection accuracy.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

This chapter provides an overview of the context and foundational concepts relevant to the research presented in this thesis. It outlines the primary challenges within the field, exploring the underlying factors that give rise to these issues. Additionally, we discuss our approach and key contributions aimed at addressing these challenges. Finally, we offer a brief summary of the organization of the thesis, guiding the reader through the structure of the upcoming chapters.

1.1 Background

Since its debut in 2017, the Transformer [48] has revolutionized a wide range of NLP tasks and has swiftly expanded its influence into the realm of computer vision, proving instrumental in tasks such as image recognition [6, 10, 12, 17, 35, 47] and object detection [5, 22, 32, 33, 38, 51, 56, 59, 61, 62]. The DEtection TRansformer (DETR) [5] stands at the forefront, being the first to adapt the Transformer's encoder-decoder architecture for the object detection task. DETR's innovation lies in its object queries, which engage with CNN-generated feature maps to concurrently predict an object's category and its spatial location. A notable feature of DETR is its ability to perform detection in an end-to-end manner, a function facilitated by the integration of Hungarian matching. Despite these advancements, DETR is hindered by suboptimal training efficiency and performance. To address these shortcomings, subsequent research has been geared toward enhancing DETR while maintaining the integrity of its original encoder-decoder architecture. Among these advancements, Deformable DETR [61]

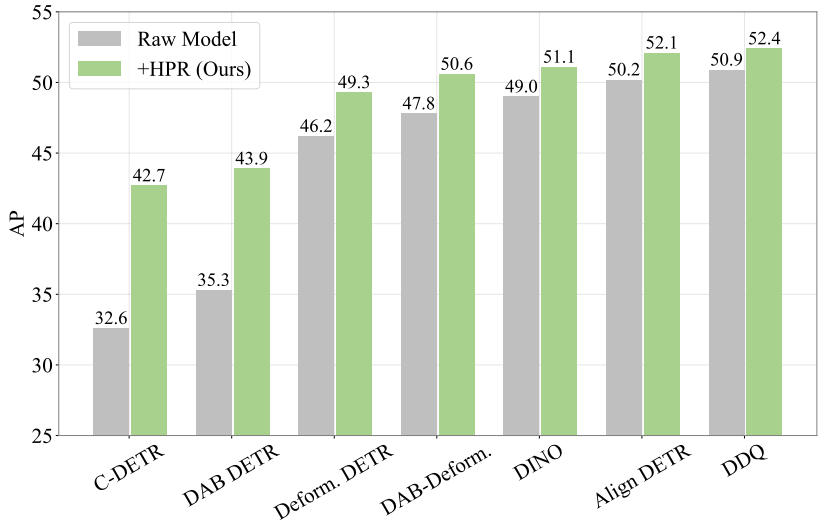

FIGURE 1.1. Applying Hybrid Proposal Refiner (HPR) to the DETR series including Conditional DETR [38], DAB DETR [33], Deformable DETR [61], DAB-Deformable DETR [33], DINO [56], Align DETR [4] and DDQ [59] on COCO dataset. All models use a ResNet-50 backbone and a 12-epoch training schedule. For efficiency, we use 300 queries for DDQ [59] and DDQ equipped with HPR.

is a prominent example, promoting DETR's capabilities by incorporating a deformable encoder and a deformable attention mechanism.

Before the advent of DETR, Faster R-CNN [43] was commonly viewed as the seminal model for object detection. It divides the detection framework into several distinct components including the backbone network, the neck network, the Region Proposal Network (RPN), and a second-stage [19, 43] or multiple-stage [3, 7] proposal refiner. The architecture of Faster R-CNN can be described as an "RPN-refiner" setup. In this structure, the RPN initially generates a collection of object proposals. Subsequently, the proposal refiner, namely the R-CNN head, undertakes the task of categorizing each proposal and more accurately adjusting their spatial coordinates.

1.2 Motivations and Contributions

In this work, we revisit DETR series from the Faster R-CNN perspective. We posit that the *encoder-decoder* structure of the DETR series can be conceptualized as a refined version of the

RPN-refiner paradigm utilized by the Faster R-CNN. we select Deformable DETR [61] with a ResNet-50 backbone as our primary model due to its widespread acclaim and exceptional performance. As shown in Table 3.1, we systematically adapt the Faster R-CNN towards the Deformable DETR, by integrating each component of the Deformable DETR to the Faster R-CNN. These adaptations span various aspects, including RPN modification (from a class-agnostic to a class-aware RPN), revisions to the neck network (from an FPN to a more capable deformable encoder), improvements to the proposal refiner (from an R-CNN to more advanced refiners like deformable attention), an increase in the stages of refinement (from a two-stage to a multi-stage process), and a transformation in the positive sample matching approach (from an IoU-based one-to-many strategy to a one-to-one Hungarian matching method).

Our research yields three primary insights: (1) The application of the Hungarian matching to the Faster R-CNN notably impedes its performance. This decrease in performance is primarily because Hungarian matching sharpens feature map activations, which causes the RoI Align operation to extract a regional feature map that includes an excess of non-essential information. (2) Using object features extracted by the neck network instead of the R-CNN features produced by RoI Align significantly mitigates the decline in performance when using Hungarian matching. Thus, a modified version of Faster R-CNN can also enjoy the advantage of end-to-end detection. (3) The performance enhancement of Deformable DETR over Faster R-CNN can be largely attributed to its integration of advanced components, notably the proposal refiner (employing deformable attention in place of RoI Align) and the enhanced neck network (utilizing a deformable encoder rather than a traditional FPN).

So far, we have effectively adapted the Faster R-CNN framework into the Deformable DETR, and we have identified the improvement of Deformable DETR over Faster R-CNN is attributable to the more sophisticated neck network and the more advanced proposal refiner. Typically, an object detector is equipped with a single neck network, but it may utilize numerous proposal refiners. Our study delves into an array of proposal refiners, each offering a distinct approach to processing and refining object proposals generated by the RPN. More precisely, our thorough examination includes RoI Align, dynamic convolution, cross

attention, deformable attention, global attention, and object feature refinement. The empirical evidence from our experiments suggests that these object refinement mechanisms are mutually compatible and effective when used in conjunction. In light of these findings, we introduce a novel approach termed as the Hybrid Proposal Refiner (HPR), which incorporates various object refinement operators and facilitates feature interactions among them. As depicted in Figure 1.1, our HPR is versatile enough to be applied to a broad range of DETR models, yielding consistent improvements when compared to their vanilla versions.

The contributions of this work are threefold:

- We revisit the DETR series from the Faster R-CNN perspective, uncovering that the encoder-decoder structure in DETR series can be interpreted as analogous to the RPN-refiner paradigm of Faster R-CNN. We progressively transform the Faster R-CNN into the Deformable DETR (Table 3.1) and comprehensively study the key elements contributing to the improvement of Deformable DETR over Faster R-CNN.
- We conduct an extensive analysis of various proposal refinement strategies and introduce the Hybrid Proposal Refinement (HPR) technique. This innovation is compatible with many existing DETR models and consistently yields performance enhancements (Figure 1.1). Additionally, we introduce a novel data augmentation strategy termed data re-augmentation, which is particularly effective when used in conjunction with the proposed HPR.
- With a ResNet-50 backbone and a 36-epoch training schedule, our method attains an AP of 54.9 on COCO benchmark.

1.3 Thesis Outline

This thesis is organized as follows:

- Chapter 1 provides an overview of vision perception and object detection, and outlines the motivations and key contributions of this research.

- Chapter 2 provides a comprehensive review of the literature on object detection models.
- Chapter 3 examines the evolution of object detection from CNN-based models like Faster R-CNN to Transformer-based approaches such as DETR. It analyzes the key components driving performance improvements and introduces the HPR to enhance detection accuracy and efficiency.
- Chapter 4 concludes the thesis by summarizing the key findings and discussing future research directions.

CHAPTER 2

Literature Review

This chapter surveys the evolution of object detection methods, moving from efficient single-stage detectors (e.g., YOLO [42], SSD [34]) and two-stage R-CNN frameworks (e.g., Faster R-CNN [43], Cascade R-CNN [3]) to Transformer-based DETR approaches [5, 27, 32, 33, 38, 50, 51] that streamline the detection pipeline. This progression reflects the ongoing efforts to enhance detection accuracy, computational efficiency, and adaptability to diverse application scenarios. Additionally, recent advancements address challenges such as handling occlusions, scale variations, and real-time processing, paving the way for more robust and versatile object detection systems.

2.1 Single-stage Detectors

Single-stage approaches have gained popularity for their simplicity and real-time performance. YOLOs [1, 2, 26, 40–42, 49] stand as a seminal contribution in this domain, providing direct predictions for bounding boxes and class labels for each preset grid cell, eschewing a secondary stage for refining these proposals. Following YOLO, SSD [34] incorporates multi-level feature extraction to localize objects across various scales. Although early single-stage detectors are considered efficient, their performance is not on par with that of two-stage or multi-stage detectors. Recent advances in single-stage detectors include the development of RetinaNet, which addresses the imbalance between foreground and background samples through the application of Focal Loss [30]. This innovation enables RetinaNet to effectively learn from a large number of hard negative samples, enhancing the effectiveness of single-stage detectors. Building upon the success of Focal Loss, researchers have further explored

alternative approaches to improve the performance of single-stage detectors. Anchor-free algorithms [21, 23, 25, 46, 53, 60] are proposed to make the detector simpler, offering a more straightforward and flexible architecture while still achieving competitive performance compared to anchor-based approaches. Subsequently, the introduction of ATSS [58] further unifies anchor-based and anchor-free models, and addresses the challenges of positive and negative sample selection during training, thus elevating the overall efficiency.

Moreover, the integration of advanced backbone networks and feature enhancement techniques has further boosted the performance of single-stage detectors. Techniques such as Feature Pyramid Networks (FPN) [29] and the adoption of lightweight architectures like MobileNet [44] have enabled single-stage detectors to achieve a favorable balance between speed and accuracy. Additionally, recent research has incorporated attention mechanisms and context modeling to enhance detection capabilities in complex scenes. These ongoing advancements continue to refine single-stage detectors, making them increasingly robust and adaptable for a wide range of real-world applications.

2.2 R-CNN Series

R-CNN [16] and Fast R-CNN [15] algorithms have been instrumental in advancing the field of object detection. R-CNN introduced the concept of using region proposals combined with Convolutional Neural Networks (CNNs) to achieve significant improvements in detection accuracy. Building upon this foundation, Fast R-CNN enhanced the efficiency of the original R-CNN by integrating the region proposal and classification stages into a single, unified network, thereby reducing computational redundancy and speeding up the detection process. These two-stage frameworks established a robust groundwork for future innovations in object detection. For instance, Faster R-CNN [43] presents the Region Proposal Network (RPN), which generates potential Regions of Interest (RoIs) that are subsequently refined in the second stage. This integration of RPN with the detection network not only accelerates the proposal generation process but also allows for end-to-end training, further improving detection performance.

More recent advancements[3, 13, 37, 45, 52, 54, 57] have augmented the Faster R-CNN framework with novel architectures to boost detection capabilities. Cascade R-CNN [3] extends the two-stage model by incorporating a multi-stage cascade of classifiers and regressors, each stage progressively refining the detection results to handle varying object scales and improve bounding box predictions. Meanwhile, Sparse R-CNN [45] introduces a paradigm shift by replacing traditional region proposals with a set of learnable queries, significantly diminishing computational complexity and enhancing the model's ability to focus on informative regions within the image. By leveraging a sparse set of proposals, Sparse R-CNN achieves competitive performance with reduced computational overhead, making it a promising direction for real-time object detection applications. Overall, these advancements demonstrate the continuous evolution of two-stage object detection frameworks, incorporating innovative strategies to enhance both accuracy and efficiency while addressing computational challenges.

2.3 DETR Series.

DETR [5] has emerged as a prominent approach in object detection research, introducing an innovative paradigm that leverages Transformer [48] and Hungarian algorithm [24]. Primarily because it eliminates the need for numerous manually engineered components, such as Non-Maximum Suppression (NMS), a number of follow-up studies [27, 32, 33, 38, 50, 51] develop various advanced extensions. With an aim to incorporate multi-level features into the DETR framework, Deformable DETR [61] utilizes a multi-scale deformable attention mechanism, that focuses on a small set of representative points around a reference point. It has been demonstrated that the Deformable DETR outperforms the original DETR, particularly in the detection of smaller objects. Subsequent research has contributed to advancing the field with more sophisticated designs [4, 8, 9, 22, 55, 59, 62]: DINO [56] improves accuracy by introducing a novel query denoising scheme; $\mathcal{H}$-DETR [22] and Group DETR [8] present the hybrid matching strategy, which combines the original one-to-one matching with an auxiliary one-to-many matching; $\mathcal{C}$o-DETR [62] introduces a collaborative hybrid assignment training scheme; DDQ [59] suggests that queries under the one-to-one assignment should exhibit

both density and uniqueness; Align DETR [4] incorporates a localization-precision-aware classification loss into its optimization process, and introduces a prime sample weighting mechanism to suppress the interference from unimportant samples.

CHAPTER 3

Hybrid Proposal Refiner: Revisiting DETR Series from the Faster R-CNN Perspective

---

This chapter delves into the advancements in object detection by comparing the innovative DETR series with the traditional Faster R-CNN framework. It highlights how DETR leverages the Transformer architecture to achieve end-to-end detection, offering improvements through models like Deformable DETR. By systematically adapting Faster R-CNN to incorporate elements of Deformable DETR, the chapter identifies key factors that enhance performance, such as deformable attention mechanisms. Additionally, it explores various proposal refinement techniques and introduces a novel Hybrid Proposal Refiner (HPR), which synergizes multiple methods to boost detection accuracy. The integration of HPR into existing DETR models demonstrates significant performance gains, showcasing the chapter's contribution to advancing object detection technology.

3.1 Method

In Section 3.1.1, we examine the progression from Faster R-CNN to Deformable DETR, highlighting the significant advancements achieved in this transition. Building upon the insight that the *encoder-decoder* architecture of the DETR series can be conceptually interpreted as an enhanced iteration of the *RPN-refiner* framework employed by Faster R-CNN, we introduce the Hybrid Proposal Refine (HPR). Additionally, we provide a comprehensive discussion on the application of HPR to various DETR models in Section 3.1.2, demonstrating its effectiveness and versatility within different architectural contexts.

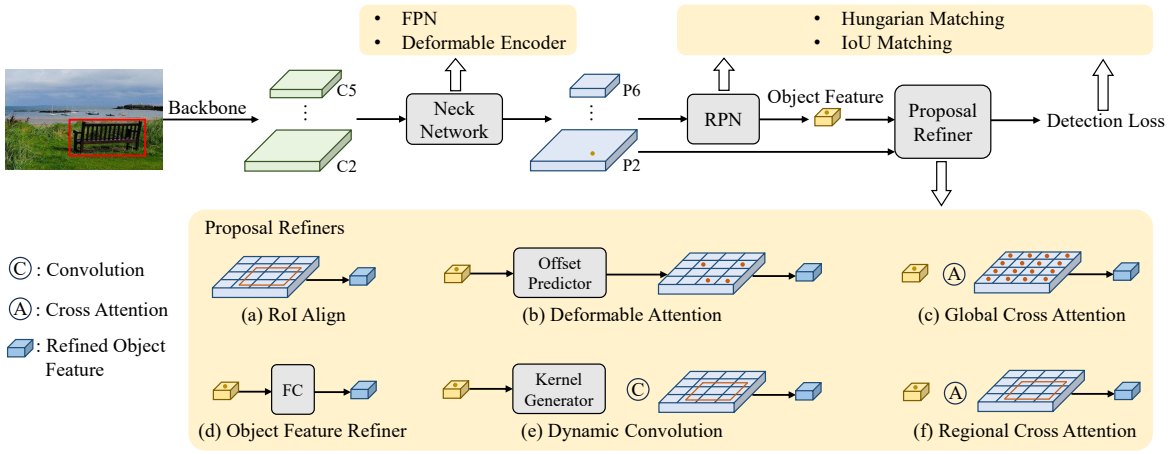

FIGURE 3.1. We regard the *encoder-decoder* structure employed by the DETR series as a refined version of the *RPN-refiner* paradigm utilized in Faster R-CNN. We investigate various elements (highlighted by yellow) that contribute to the transition from Faster R-CNN to Deformable DETR. Our HPR is predicated on exploring a multitude of proposal enhancement strategies that operate on different levels: regional (a, b, e, f), global (c), and point level (d).

3.1.1 From Faster R-CNN to Deformable DETR

As illustrated in Figure 3.1, we study a number of factors that are involved in the evolution from Faster R-CNN to Deformable DETR, including the RPN, the neck network, the proposal refiners, the stages of refinement, and the positive sample matching strategy. The performance of each intermediate modification is reported in Table 3.1.

Faster R-CNN Baseline. Our baseline is established by employing Faster R-CNN with a ResNet-50 backbone and an FPN neck network, utilizing a 12-epoch training schedule. We adopt RoI Align to extract region features. This configuration achieves a 36.5 AP on the COCO `val` set.

Class-Agnostic RPN vs. Class-Aware RPN. We modify the class-agnostic RPN used in Faster-RCNN to be class-aware, in line with the approach taken by Deformable DETR. This results in a slight drop in performance, from 36.5 to 36.1.

TABLE 3.1. Step by step, we transform the Faster R-CNN [43] into the Deformable DETR [61]. We report AP on COCO benchmark. Object feature denotes RPN's point feature extracted by the neck network. Refer to Section 3.1.1 for more details.

Model	AP
Faster R-CNN (ResNet-50, FPN, 12-epoch)	36.5
+ Class-Agnostic RPN→Class-Aware RPN	36.1 (-0.4)
+ FPN→Deformable Encoder	44.0 (+7.9)
+ IoU Matching→Hungarian Matching (RPN)	32.7 (-11.3)
+ IoU Matching→Hungarian Matching (R-CNN)	32.2 (-0.5)
+ RoI Feature→Object Feature	41.2 (+9.0)
+ Object Feat.→Object Feat. + RoI Feat.	41.7 (+0.5)
+ Object Feat. + RoI Feat.→Deformable Attention	44.2 (+2.5)
+ 6× Deformable Attention	46.2 (+2.0)

Neck Network. Deformable DETR utilizes a powerful neck network known as the deformable encoder. Transformation from an FPN-style neck network to a deformable encoder enhances the AP from 36.1 to 44.0.

IoU Matching (One-to-Many) vs. Hungarian Matching (One-to-One). One of the most appealing advantages of the DETR series is its capability for end-to-end detection. This is attributed to the utilization of Hungarian matching, as opposed to the long-standing IoU-based matching strategy employed by the Faster R-CNN series. As shown in Table 3.1, transformation from IoU-based matching to Hungarian matching for RPN dramatically hinders the detector, resulting in a significant AP drop from 44.0 to 32.7 for this Faster R-CNN variant. In addition, the application of Hungarian matching in R-CNN yields an AP degradation of 0.5. We conjecture that the first performance drop (44.0→32.7) arises from the RoI Align operator. The use of Hungarian matching intensifies the feature map activations; however, the RoI Align operator extracts a regional feature map that includes an excess of non-essential information. To verify our hypothesis, we conduct two studies. First, we present visualizations of two activation maps in Figure 3.2: one map generated by a Faster R-CNN variant that uses Hungarian matching in the RPN, and another produced by a different Faster R-CNN variant that employs IoU matching in the RPN. It can be seen that the activation map of the former is much sharper than that of the latter. Next, we simply retrain a class-aware

TABLE 3.2. Step by step, we transform the Faster R-CNN [43] into the Deformable DETR [61]. We report AP on COCO benchmark. Object feature denotes RPN's point feature extracted by the neck network. Refer to Section 3.1.1 for more details.

Matching Strategy	AP	AP_l	AP_m	AP_s
IoU	38.3	51.2	43.2	21.1
Hungarian	38.4	48.6	42.2	24.2

RPN with deformable encoder and the application of Hungarian matching, which can be viewed as a single-stage detector. The results presented in Table 3.2 show that Hungarian matching does not impede the performance of this enhanced RPN, suggesting that the object features (i.e., point features) utilized by the RPN are sufficient for object localization and classification.

Both quantitative and qualitative results verify our hypothesis—the RoI Align operator extracts a regional feature map that includes an excess of non-essential information when introducing Hungarian matching into the Faster R-CNN.

RoI Feature vs. Object Feature. Based on the aforementioned observation, we utilize the structure of "object feature→FC" as the second stage proposal refiner instead of the

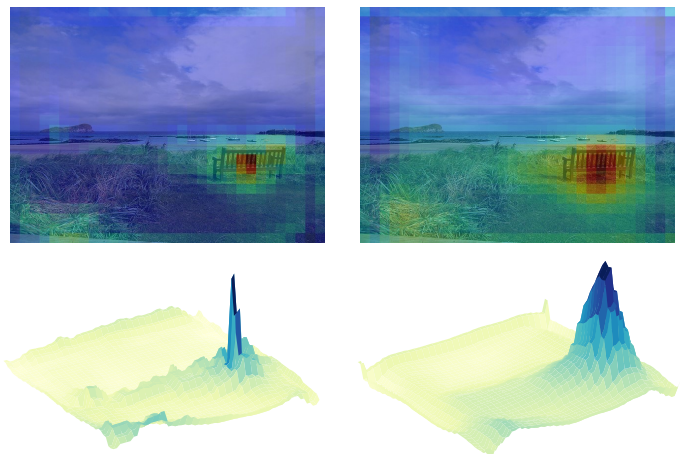

FIGURE 3.2. Visualization of two activation maps generated by variants of Faster R-CNN using either Hungarian matching or IoU matching.

structure of "RoI Align→region feature→CNN→FC" (R-CNN), yielding a significant AP improvement from 32.2 to 41.2. Additionally, as shown in Table 3.1, integrating the RoI features into the object features further results in an AP enhancement of 0.5. These studies indicate that a proposal refinement module, more appropriate than RoI Align, aligns effectively with the application of Hungarian matching.

More Powerful Proposal Refiner. Deformable DETR introduces a deformable attention mechanism, which enhances object features by incorporating the features derived from a set of representative points. We replace the proposal refiner from "object feature + RoI feature→FC" to a deformable decoder, which introduces a sophisticated interaction between each object feature and the feature maps extracted by the neck network (i.e., deformable encoder). As shown in Table 3.1, this alteration leads to a +2.5 AP improvement. Finally, by adopting $6\times$ deformable decoders, we achieve an AP of 46.2, marking the successful transition from Faster R-CNN to Deformable DETR.

3.1.2 Hybrid Proposal Refiner

We have identified that the improvement of Deformable DETR over Faster R-CNN can be credited to its sophisticated neck network and its advanced proposal refiner. In general, an object detector is equipped with a single neck network, yet it may utilize numerous proposal refiners. As illustrated in Figure 3.1, prior to presenting our HPR, we first explore other potential proposal refiners besides RoI Align (Figure 3.1a) and deformable attention (Figure 3.1b).

Notations. Let H and W represent the height and width of the input image, respectively. We denote the feature maps extracted by a backbone network as $\{\mathcal{C}_l \in \mathbb{R}^{H/2^l \times W/2^l \times d_l}\}$, where d_l is the feature dimension and l denotes the stage number. The feature maps encoded by the neck network[1] are denoted as $\{\mathcal{P}_l \in \mathbb{R}^{H/2^l \times W/2^l \times D}\}$, where D is the feature dimension. We use $\boldsymbol{p}_i \in \mathbb{R}^D$ to denote the object feature (i.e., point feature used by RPN) of the i-th

[1]In this work, all modules designed to enhance the features produced by the backbone network are collectively referred to as the neck network. This includes the FPN, the Transformer-encoder, and the deformable encoder.

object proposal with bounding box $\boldsymbol{b}_i = (x_i,\ y_i,\ w_i,\ h_i)$. We represent the RoI feature of $\boldsymbol{b}_i$ as $\boldsymbol{r}_i \in \mathbb{R}^{7\times7\times D}$. $\boldsymbol{r}_i$ is generated by the RoI Align operator.

Global Cross Attention (Figure 3.1c). This mechanism is adopted by the original DETR [5], where a set of learnable object queries are introduced to gather information from $\mathcal{P}_5$ via cross attention operation. Note that using global attention is computationally expensive.

Object Feature Refinement (Figure 3.1d). In Section 3.1.1, this strategy has been discussed in the evolution from Faster R-CNN to Deformable DETR. The object feature refiner directly processes the object features $\{\boldsymbol{p}_i\}$ to refine the proposals generated by RPN.

Dynamic Convolution (Figure 3.1e). Dynamic convolution [45] enhances object features by by facilitating the interaction between each object feature $\boldsymbol{p}_i$ and the corresponding RoI feature $\boldsymbol{r}_i$. Specifically, $\boldsymbol{p}_i$ first undergoes processing by FC layers to generate convolutional kernels. Subsequently, these kernels are applied to $\boldsymbol{r}_i$ through convolution layers followed by FC layers, resulting in an enhanced object feature of $\boldsymbol{p}_i$.

Regional Cross Attention (Figure 3.1f). An alternative to perform the interaction between the object feature $\boldsymbol{p}_i$ and its RoI feature $\boldsymbol{r}_i$ is to adopt cross attention, where $\boldsymbol{p}_i$ serves as the query while the elements in $\boldsymbol{r}_i$ act as the keys and values. In this work, we refer to this object feature refinement strategy as regional cross attention.

Hybrid Proposal Refiner (HPR). Up to this point, we have explored various strategies aimed at refining proposals, which operate on different levels: global (global cross attention), regional (RoI Align, deformable attention, dynamic convolution and regional cross attention) and point level (object feature refinement). As described in Section 3.1.1, it has been noted that the RoI Align operation does not effectively coincide with Hungarian matching algorithm. In addition, we observe that the interaction between object features and the corresponding regional features is essential for the effectiveness of high-performance end-to-end detectors.

Unlike previous DETR models that merely include one proposal refiner, our HPR integrates the strengths of various regional proposal refinement techniques such as deformable attention, dynamic convolution, and regional cross attention. Even though these regional proposal

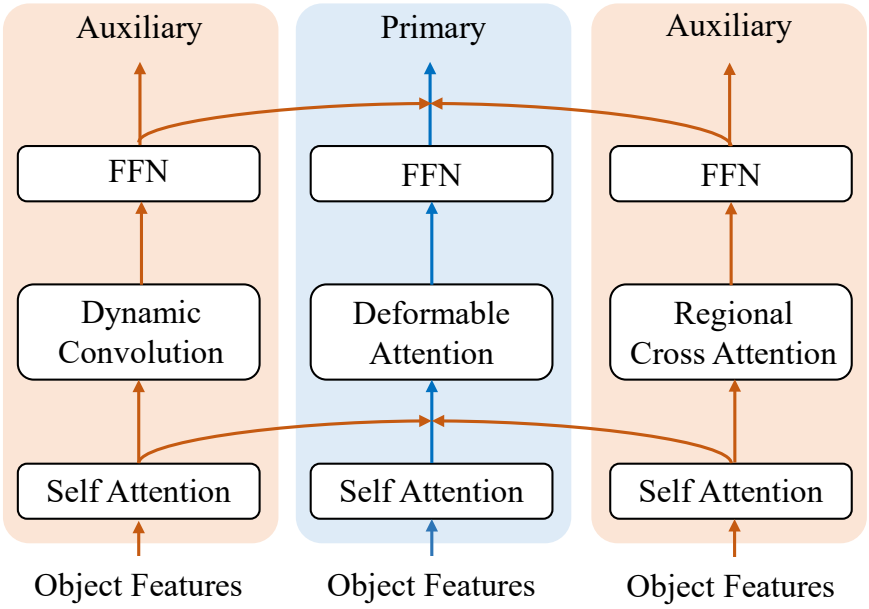

FIGURE 3.3. Illustration of the HPR module. The auxiliary refiners inject implicit information into the intermediate features of the primary refiner. We use $6\times$ HPRs by default.

refiners are designed to capture the most essential features of foreground objects, the methods they use to encode local features vary significantly. Deformable attention adopts a sparse set of point features. In contrast, both dynamic convolution and regional cross attention employ RoI features, but they differ in their utilization of object features: the former kernelizes the object features, while the latter regards the object features as the queries of the cross attention.

As shown in Figure 3.3, to take full advantage of the potential of each proposal refiner, HPR designates one refiner to function as the primary refiner, while the others act as the auxiliary. The auxiliary refiners inject implicit information into the intermediate features of the primary refiner. Specifically, we integrate the self-attention and FFN features from the auxiliary proposal refiner into their counterparts in the primary proposal refiner using a simple addition operator with learnable weights. In Section 3.2.2, we also investigate other alternatives for information integration. Note that each proposal refiner is supervised by an independent detection loss. The loss weights of the primary proposal refiner and two auxiliary refiners are

set to 1.0, 0.5 and 0.5, respectively. We use the same loss function as the one employed in Deformable DETR.

Application of HPR to DETR Series. Like most DETR detectors, stacking multiple HPRs is feasible to enhance overall performance. By default, we stack 6 HPRs. Our HPR can be incorporated into various DETR detectors that only have a single proposal refiner by appending the auxiliary refiners to the primary one. Figure 1.1 demonstrates the consistent performance improvement.

Data Re-Augmentation. We also introduce a novel data augmentation strategy termed "data re-augmentation", which first copies data that has been augmented by normal augmentation and then applies strong augmentations, including color jitter and geometric transformations, to the copies. This yields a new training batch that contains both normally augmented images and strongly augmented images. Our data re-augmentation technique differs from batch augmentation [20] in two aspects: (1) it copies weakly augmented images rather than the raw images; (2) it applies distinct and stronger augmentations to the copies. We experimentally find this novel augmentation works well with our HPR.

3.2 Experiments

Dataset and Evaluation Metric. We conduct experiments on COCO [31] benchmark. It offers 118,287 labeled images across 80 object categories in its `train` set. The `val` set consists of 5,000 images. Following common practice, we report average precision (AP) on COCO `val` split.

Implementation Details. Our code base is built upon MMDetection [5]. Unless otherwise specified, we adopt ResNet-50 [18] pre-trained on ImageNet-1K [11] as the backbone under a 12-epoch training schedule. By default, 900 object queries are adopted. We use the AdamW [36] optimizer with a learning rate of $1e{-}4$. We adopt DETR-style normal data augmentation following [4, 8, 22, 56, 62] and the proposed data re-augmentation technique. When compared with other approaches, we utilize a larger backbone (Swin-L [35] pre-trained

on ImageNet-22K [11]) and longer training schedules (24 or 36 epochs), and incorporate large-scale jitter with copy-paste technique [14] into the normal data augmentation.

TABLE 3.3. Comparison with state-of-the-art DETR models on the COCO `val` set utilizing a ResNet-50 backbone. Considering GPU memory utilization, we use 300 queries when applying HPR to DDQ [59]. †: the application of large-scale jitter data augmentation.

Method	Backbone	#Queries	#Epochs	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
Conditional DETR [38]	ResNet-50	300	108	43.0	64.0	45.7	22.7	46.7	61.5
Anchor DETR [51]		300	50	42.1	63.1	44.9	22.3	46.2	60.0
Efficient DETR [55]		300	50	45.1	63.1	49.1	28.3	48.4	59.0
DAB DETR [33]		900	50	45.7	66.2	49.0	26.1	49.4	63.1
Deformable DETR [61]		300	50	46.9	65.6	51.0	29.6	50.1	61.6
DN-Deformable DETR [27]		900	50	48.6	67.4	52.7	31.0	52.0	63.7
$\mathcal{H}$-Deformable DETR [22]		300	12	48.7	66.4	52.9	31.2	51.5	63.5
$\mathcal{H}$-Deformable DETR [22]		300	36	50.0	-	-	32.9	52.7	65.3
DINO [56]		900	12	49.4	66.9	53.8	32.3	52.5	63.9
DINO [56]		900	36	51.2	69.0	55.8	35.0	54.3	65.3
Group DETR [8]		900	12	50.1	-	-	32.4	53.2	64.7
Align DETR [4]		900	12	50.2	67.8	54.4	32.9	53.3	65.0
Align DETR [4]		900	24	51.3	68.2	56.1	35.5	55.1	65.6
DETA [39]		900	12	50.5	67.6	55.3	33.1	54.7	65.2
DETA [39]		900	24	51.6	69.0	56.7	34.0	55.8	66.5
DDQ [59]		900	12	51.3	68.6	56.4	33.5	54.9	65.9
DDQ [59]		900	24	52.0	69.5	57.2	35.2	54.9	65.9
$\mathcal{C}$o-Deformable DETR [62]		900	12	49.5	67.6	54.3	32.4	52.7	63.7
$\mathcal{C}$o-DINO DETR [62]		900	12	52.1	69.4	57.1	35.4	55.4	65.9
Deformable DETR with HPR		900	12	50.6	68.7	55.5	34.4	53.9	63.5
Deformable DETR with HPR		900	24	51.9	70.0	57.0	35.3	55.0	65.3
DINO with HPR		900	12	51.1	68.6	55.7	34.6	54.5	64.9
DINO with HPR		900	24	51.9	69.7	56.8	34.9	55.0	65.8
Align DETR with HPR		900	12	52.1	69.6	56.9	35.6	55.4	66.6
Align DETR with HPR		900	24	52.7	69.8	57.2	35.8	56.0	66.4
Align DETR with HPR†		900	12	52.4	70.3	57.2	35.9	56.3	68.5
Align DETR with HPR†		900	24	54.2	72.1	58.8	37.8	57.9	70.0
DDQ with HPR		300	12	52.4	69.9	57.5	35.9	55.5	66.7
DDQ with HPR		300	24	52.5	69.8	57.6	35.4	55.5	67.0
DDQ with HPR†		300	12	53.0	70.6	58.0	35.3	56.3	68.6
DDQ with HPR†		300	24	54.2	72.0	59.6	37.3	57.8	69.1
DDQ with HPR†		300	36	**54.9**	**72.4**	**60.3**	**37.7**	**58.9**	**69.6**

TABLE 3.4. Comparison with other DETR models on the COCO `val` set utilizing a Swin-L backbone pre-trained on ImageNet-22K. Considering GPU memory utilization, we use 300 queries when applying HPR to DDQ [59]. †: the utilization of large-scale jitter.

Method	Backbone	#Queries	#Epochs	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
HTC [7]	Swin-L (IN-22K)	900	36	57.1	75.6	62.5	42.4	60.7	71.1
Group-DINO [8]		900	36	58.4	-	-	41.0	62.5	73.9
DETA [39]		900	24	58.5	76.5	64.4	38.5	62.6	73.8
DINO [56]		900	12	57.5	-	-	-	-	-
DINO [56]		900	36	58.5	77.0	64.1	41.5	62.3	74.0
DDQ [59]		900	36	58.7	76.8	64.5	41.6	62.9	74.3
Mask DINO [28]		300	50	59.0	-	-	-	-	-
$\mathcal{H}$-Deformable DETR [22]		900	12	55.9	-	-	39.1	59.9	72.2
$\mathcal{H}$-Deformable DETR [22]		900	36	57.1	-	-	39.7	61.4	73.4
$\mathcal{H}$-DINO [22]		900	36	59.4	77.8	65.4	43.1	63.1	74.2
DDQ with HPR		300	12	58.7	76.7	64.5	41.5	62.5	74.6
DDQ with HPR†		300	12	58.4	76.8	64.3	41.2	62.5	75.1
DDQ with HPR†		300	24	59.3	77.6	65.0	43.1	63.4	75.5
AlignDETR with HPR		900	12	58.6	76.8	64.0	40.9	62.7	75.4
AlignDETR with HPR		900	24	59.3	77.5	64.7	41.9	63.7	75.2
AlignDETR with HPR†		900	12	58.5	76.7	63.7	41.6	62.8	76.6
AlignDETR with HPR†		900	24	59.6	77.9	64.5	42.6	64.0	**76.9**
AlignDETR with HPR†		900	36	**60.0**	**78.0**	**65.5**	**43.8**	**64.5**	76.6

3.2.1 Main Results

As shown in Figure 1.1, our HPR can be applied to various DETR detectors, including Conditional DETR [38], DAB DETR [33], Deformable DETR [61], DAB-Deformable DETR [33], DINO [56], Align DETR [4] and DDQ [59]. Models equipped with our HPR technique consistently outperform their counterparts without HPR, showing improvements ranging from +1.5 to +10.1 AP.

The comparison with state-of-the-art methods utilizing a ResNet-50 backbone is presented in Table 3.3. Notably, by applying HPR to a strong DETR, namely DDQ [59], we achieve an AP of 54.9 under a 36-epoch training schedule. In Table 3.4, we compare our method with other approaches using a Swin-L backbone. When applied to DDQ [59], and AlignDETR [4], our approach achieves AP scores of 59.3 and 60.0, respectively.

TABLE 3.5. Performance of each proposal refiner.

Proposal Refiner	AP	AP_l	AP_m	AP_s
Global Cross Attention	42.3	56.3	45.4	26.8
RoI Align	32.2	37.3	36.9	23.6
Deformable Attention	47.8	62.0	51.2	30.6
Dynamic Convolution	48.3	62.7	51.2	32.3
Regional Cross Attention	47.6	61.5	50.6	31.7
Object Feature Refiner	41.2	51.7	44.8	26.9

3.2.2 Ablation Studies

Unless otherwise specified, for all ablation studies, an enhanced Deformable DETR introduced by DINO [56] serves as our base model. It achieves 47.8 AP, using a ResNet-50 backbone, normal data augmentation and 300 object queries under a 12-epoch training schedule.

Various Proposal Refiners. In Section 3.1.2, we introduce a variety of proposal refiners that function at distinct levels: global (global cross attention), regional (RoI Align, deformable attention, dynamic convolution and regional cross attention) and point level (object feature refiner). The performance of each refiner is detailed in Table 3.5. With the exception of RoI Align and the object feature refiner, all refiners utilize a six-stage refinement process. As explored in Section 3.1.1, the RoI Align technique does not effectively integrate with Hungarian matching, leading to suboptimal results. The global cross attention mechanism, as proposed by the original DETR, incurs significant computational overhead and poses challenges for the integration of multi-level feature maps that modern DETR detectors typically need. In contrast to the simple object feature refiner which adopts a single FC layer for object feature enhancement, deformable attention (DA), dynamic convolution (DC) and regional cross attention (RCA) exhibit superior performance. This improvement stems from their intricate architectures, which facilitate interactions between object and regional features. Thus, DA, DC and RCA are adopted in our HPR.

Feature Integration. As shown in Figure 3.3, the self-attention (SA) and feed-forward network (FFN) features from the auxiliary proposal refiners are integrated into their counterparts within the primary proposal refiner. Each refiner is composed of a SA layer, a dedicated

TABLE 3.6. Ablation study on the integration of various features.

SA	Dedicated Module	FFN	AP
✓			48.0
	✓		46.5
		✓	49.2
✓	✓		48.6
	✓	✓	48.6
✓		✓	**49.3**
✓	✓	✓	49.1

TABLE 3.7. Ablation study on the integration weights.

Weight	Type	Initialization	AP
Fixed	Scalar	1:1:1	48.9
Fixed	Scalar	2:1:1	49.1
Learnable	Scalar	1:1:1	48.9
Learnable	Scalar	2:1:1	48.8
Learnable	Vector	1:1:1	**49.3**
Learnable	Vector	2:1:1	49.0

module (deformable attention, dynamic convolution or regional cross attention), and a FFN layer. In Table 3.6, we study the effectiveness of different features for information injection, including SA features, FFN features, and features from the dedicated module. Experimentally, we find that injecting SA features and FFN features into the primary proposal refiner yields the best performance.

In addition, we examine the integration weights. Let $\boldsymbol{f}_p$, $\boldsymbol{f}_{a1}$ and $\boldsymbol{f}_{a2}$ denote the features extracted by the FFN or SA layer of the primary proposal refiner, the first auxiliary proposal refiner, and the second auxiliary proposal refiner, respectively. The corresponding refined feature $\boldsymbol{f}'_p$ is computed as $\boldsymbol{f}'_p = w_p\boldsymbol{f}_p + w_{a1}\boldsymbol{f}_{a1} + w_{a2}\boldsymbol{f}_{a2}$. In Table 3.7, we study several factors including: (1) whether these weights $\{w_p, w_{a1}, w_{a2}\}$ are fixed or learnable; (2) the data type of $\{w_p, w_{a1}, w_{a2}\}$ as either scalar or vector of the same dimension of $\boldsymbol{f}_p/\boldsymbol{f}_{a1}/\boldsymbol{f}_{a2}$; (3) the initial values of $\{w_p, w_{a1}, w_{a2}\}$.

Performance Enhancement. We introduce data re-augmentation in the end of Section 3.1.2. Table 3.8 shows the effects of incorporating data re-augmentation and increasing the number

TABLE 3.8. Study on data re-augmentation and more object queries.

HPR	Data Re-Augmentation	900 Queries	AP
			47.8
✓			49.3
✓		✓	49.8
✓	✓		50.3
✓	✓	✓	**50.6**

TABLE 3.9. Comparison among standard data augmentation (the first and second rows), batch augmentation [20] (the third row), and data re-augmentation (the last row).

Augmentation Strategy	AP
Normal Augmentation	49.3
Strong Augmentation	48.4
Batch Augmentation	49.6
Data Re-Augmentation	**50.3**

TABLE 3.10. Study on the integration of auxiliary proposal refiners (dynamic convolution and regional cross attention) into the primary proposal refiner (deformable attention). Refer to the supplementary materials for more results.

Deformable Att.	Dynamic Conv.	Regional CA	AP
✓			47.8
✓	✓		48.9
✓		✓	48.4
✓	✓	✓	**49.3**

of object queries from 300 to 900. Our data re-augmentation involves first duplicating data that has undergone normal augmentation, and then applying strong augmentations to these duplicates to create a new batch. The training is conducted on the combination of the original batch (augmented by normal augmentation) and the new batch (augmented by data re-augmentation). To evaluate the effectiveness of our data re-augmentation, we compare it with the standard normal and strong augmentation, and the batch augmentation [20] in Table 3.9.

Integration of Auxiliary Object Refiners into Primary Object Refiner. We adopt deformable attention as our primary proposal refiner. The performance improvements achieved

TABLE 3.11. Ablation study on primary object refiners. Att.: attention. CA: cross attention. Conv.: convolution.

Primary	Auxiliary-1	Auxiliary-2	AP
Deformable Att.	-	-	47.8
Deformable Att.	Deformable Att.	Deformable Att.	48.5
Regional CA	Dynamic Conv.	Deformable Att.	48.9
Dynamic Conv.	Regional CA	Deformable Att.	48.8
Deformable Att.	Regional CA	Dynamic Conv.	**49.3**

TABLE 3.12. Ablation study on loss weight.

Loss Weight	AP	AP_l	AP_m	AP_s
1:1:1	49.1	63.8	51.7	32.5
2:1:1	**49.3**	62.8	52.4	32.6

by employing dynamic convolution, regional cross attention, and their combination as the auxiliary refiners are presented in Table 3.10. The inclusion of each auxiliary refiner enhances the effectiveness of using a solitary primary refiner.

Ablation Study on Primary Object Refiners. Figure 3.3 illustrates a scenario in which deformable attention is employed as the primary refiner, supported by dynamic convolution and regional cross attention as auxiliary refiners. In Table 3.11, we delve into alternative configurations, assigning the roles of primary refiners to both regional cross attention and dynamic convolution separately. We compare these setups against the original arrangement where deformable attention is the primary refiner. Additionally, we establish a baseline that utilizes deformable attention for the primary refiner, along with two auxiliary refiners. Our HPR amalgamates the strengths of diverse regional proposal refinement techniques, thereby surpassing the baseline that employs a singular type of proposal refinement strategy.

Ablation Study on Loss Weight. We perform an ablation study to examine how different loss weights between the primary and auxiliary refiners affect performance. Table 3.12 shows that our model achieves an AP of 49.3 under a loss weight distribution of 2:1:1.

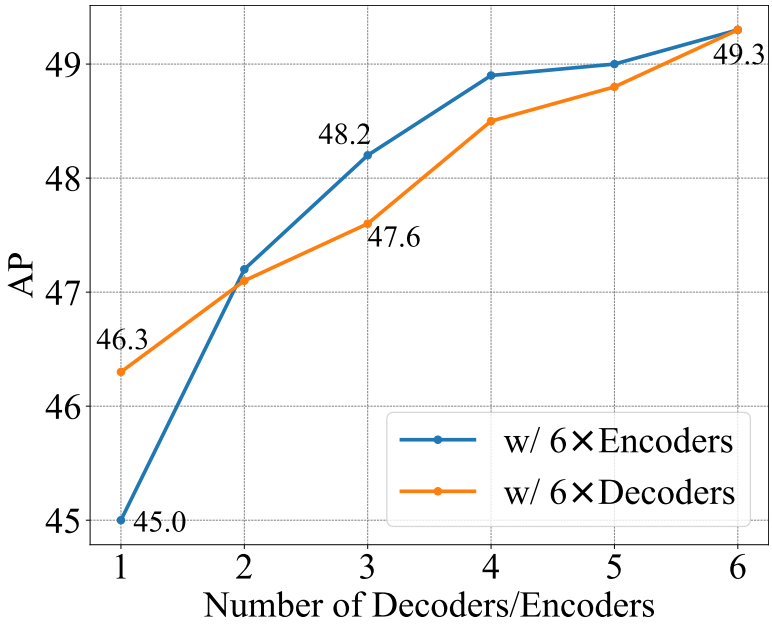

FIGURE 3.4. Ablation study on variations in the number of encoders (deformable encoders) and decoders (HPRs). Blue line: variation in the number of decoders within a model with 6× encoders. Orange line: variation in the number of encoders within a model with 6× decoders.

Examination of Encoder and Decoder Number Variations. We explore the impact of varying the number of encoders and decoders on system performance. The number of decoders varies from 1 to 6 in a model with 6× encoders. Similarly, we apply this variation to a model with 6× decoders. The results are presented in Figure 3.4.

Operational Mechanisms of Various Proposal Refinement Strategies. In the realm of object feature utilization, the operational mechanisms of deformable attention, dynamic convolution, and regional cross attention exhibit distinct characteristics. Deformable attention predicts a sparse set of point features corresponding to each specific object feature. In contrast, dynamic convolution transforms object features into kernels—the generated kernels then slide over the RoI features to yield enhanced object features. Regional cross attention, meanwhile, operates by integrating object features with RoI features via a cross attention mechanism, wherein object features are treated as queries and RoI features as keys. In Figure 3.5, we visualize the activation maps for the three proposal refiners. It is evident that each refiner focuses on different areas and semantics of the object.

FIGURE 3.5. Visualizations of the activation maps for deformable attention (the second row), dynamic convolution (the third row), and regional cross attention (the last row).

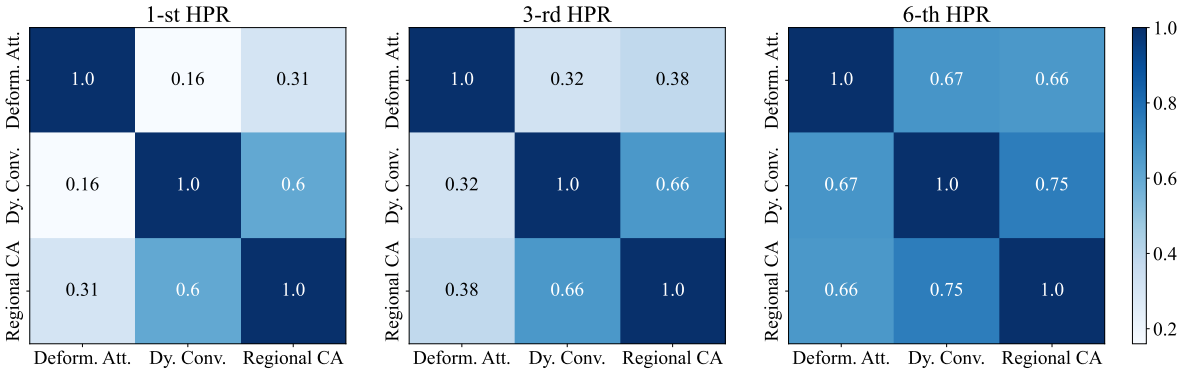

FIGURE 3.6. Visualizations for cosine similarities of various proposal refiners in distinct HPR stages.

Additionally, we gather statistics on the cosine similarities of features extracted by two proposal refiners across object queries and throughout the images from the COCO `val` set. These statistics enable us to calculate an average cosine similarity s, which serves as a measure of the resemblance between the features extracted by the two refiners. A greater value of s

FIGURE 3.7. Visualizations of the activation maps generated by variants of Faster R-CNN using either IoU matching (the second row) or Hungarian matching (the third row).

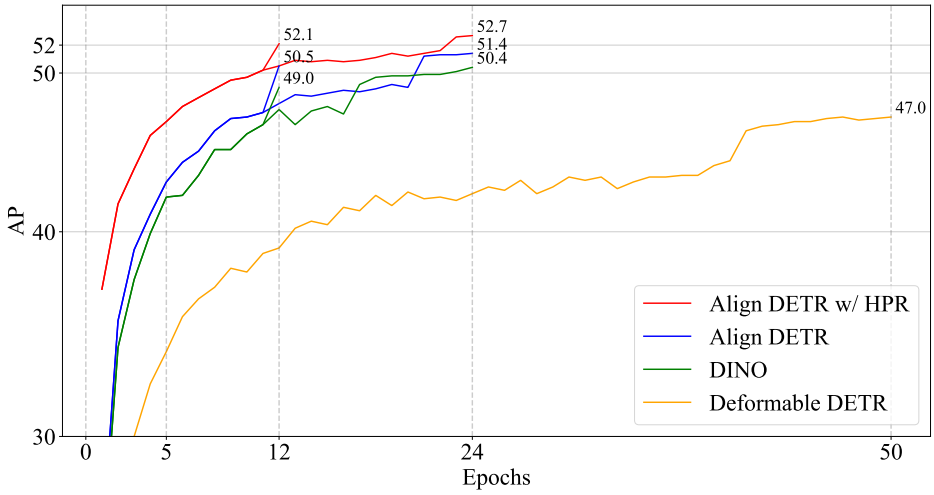

FIGURE 3.8. Training curves for AlignDETR equipped with our HPR, the original AlignDETR, DINO, and Deformable DETR.

suggests a higher degree of similarity in the features extracted by these refiners. We use the features derived from the first, the intermediate (third stage), and the last stages of HPR for similarity calculation. The visualizations are presented in Figure 3.6. It is evident that during

TABLE 3.13. Ablation study on data re-augmentation and large-scale jitter (LSJ) augmentation.

#Epochs	Data Re-aug.	LSJ	AP
12			49.3
	✓		50.3
		✓	49.3
	✓	✓	50.4
24			50.5
	✓		51.3
		✓	51.6
	✓	✓	52.8

the preliminary stages, specifically the first and third stages, the features encoded by various refiners exhibit considerable variance. In contrast, in the last stage, there is a notable increase in feature similarity, which is attributed to their convergence within a common latent space.

Qualitative Study on Positive Sample Matching Strategies. In Figure 3.2, we visualize two activation maps generated by variants of Faster R-CNN using either Hungarian matching or IoU matching. We show more visualizations in Figure 3.7.

Training Curve Analysis. Figure 3.8 illustrates a comparative analysis of the training progression for the Align DETR [4] equipped with our HPR, alongside its original version and two other DETR variations, namely DINO [56] and Deformable DETR [61]. The incorporation of our HPR significantly accelerates the training convergence.

Ablation Study on Data Augmentations. We verify the effects of the proposed data re-augmentation and large-scale jitter augmentation in Table 3.13. It is observed that these two data augmentation strategies demonstrate compatibility in their application.

Distinction with $\mathcal{C}$o-DETR. The underlying motivations of these approaches diverge significantly. Our HPR is developed through the evolution from Faster R-CNN [43] to DETR [5], incorporating the unique characteristics of various proposal refiners (PRs) in object modeling. HPR integrates diverse features extracted by multiple types of PRs, whereas $\mathcal{C}$o-DETR [62]

Method	GPU hours	GFLOPs	FPS	AP
Deform. DETR	120	271	10.1	47.8
Ours (w/ Deform. DETR)	152	312	8.3	50.6
DINO	132	271	10.1	49.0
Ours (w/ DINO)	184	312	8.3	51.1
DDQ	200	274	8.3	51.3
Ours (w/ DDQ)	304	317	6.6	53.0

TABLE 3.14. Training: 12-epoch; 8 $\times$ Nvidia V100 GPUs. Inference: single Nvidia V100 GPU; image resolution of 800×1333.

employs a singular form of PR (deformable attention). The former introduces two auxiliary one-to-many matching branches, while the latter focuses on aligning classification and regression.

Training & Inference Cost. As shown in Table 3.14, our method boosts performance by +1.7 to +2.8 AP across all baselines with modest inference overhead. The DDQ [59] variant achieves 53.0 AP, demonstrating effective accuracy-computation trade-offs under standard evaluation settings

3.3 Formulation of Proposal Refiners.

As described in Section 3.1.2 of the main paper, we use $\{\mathcal{P}_l\}$ to denote the feature maps encoded by the neck network (deformable encoder). Let $\boldsymbol{b}_i$ represent the i-th bounding box generated by the RPN. We use $\boldsymbol{p}_i$ and $\boldsymbol{r}_i$ to denote its object feature (point feature) and RoI feature, respectively. The enhanced object feature is represented by $\boldsymbol{p}'_i$. Below, we provide a formal formulation for each object refiner. For the sake of simplicity, we omit activation layers in our formulations.

R-CNN. It [43] employs a stack of convolutional layers to refine the RoI features $\{\boldsymbol{r}_i\}$. This process can be formulated as:

$$\boldsymbol{p}'_i = \mathrm{FC}(\mathrm{Conv}(\boldsymbol{r}_i)).$$

Object Feature Refiner. This strategy directly processes the object features $\{\boldsymbol{p}_i\}$ using several FC layers:

$$\boldsymbol{p}_i' = \mathrm{FC}(\boldsymbol{p}_i).$$

Dynamic Convolution. This strategy facilitates interaction between $\boldsymbol{p}_i$ and $\boldsymbol{r}_i$. $\boldsymbol{p}_i$ is first used to generate convolution kernels through FC layers, and the convolution is subsequently applied to $\boldsymbol{r}_i$. The formulation is presented as follows:

$$\boldsymbol{K}_1 = \mathrm{FC}(\boldsymbol{p}_i),$$

$$\boldsymbol{K}_2 = \mathrm{FC}(\boldsymbol{p}_i),$$

$$\boldsymbol{p}_i' = \mathrm{FC}(\mathrm{Conv}_{\boldsymbol{K}_2}(\mathrm{Conv}_{\boldsymbol{K}_1}(\boldsymbol{r}_i))),$$

where $\mathrm{Conv}_{\boldsymbol{K}}$ denotes the convolution operator with kernel $\boldsymbol{K}$.

Regional Cross Attention. It applies cross attention between $\boldsymbol{p}_i$ and $\boldsymbol{r}_i$. $\boldsymbol{p}_i$ and $\boldsymbol{r}_i$ serve as queries and keys, respectively. We formulate the process as follows:

$$\hat{\boldsymbol{p}}_i^m = \mathrm{FC}_m(\boldsymbol{p}_i),\ 1 \leq m \leq 5$$

$$\{\boldsymbol{p}_i'^m\}_{m=1}^5 = \mathrm{CrossAttention}(\{\hat{\boldsymbol{p}}_i^m\}_{m=1}^5, \boldsymbol{r}_i),$$

$$\boldsymbol{p}_i' = \mathrm{Concatenation}(\{\boldsymbol{p}_i'^m\}_{m=1}^5).$$

Deformable Attention. It uses several linear layers to predict a set of reference points with offsets Δ and the corresponding attention weights $\boldsymbol{A}$ for each $\boldsymbol{p}_i$. The entire process can be formulated as:

$$\Delta = \mathrm{FC}(\boldsymbol{p}_i),$$

$$\boldsymbol{A} = \mathrm{FC}(\boldsymbol{p}_i),$$

$$\boldsymbol{p}_i' = \mathrm{DeformableAttention}(\{\mathcal{P}_l\}, \Delta, \boldsymbol{b}_i, \boldsymbol{A}),$$

where $\mathcal{P}_l$ denotes the l-th feature map generated by the deformable encoder and $\boldsymbol{b}_i$ represents the bounding box associated with $\boldsymbol{p_i}$.

Global Cross Attention. For this mechanism, each object feature $\boldsymbol{p}_i$ (query) interacts with $\mathcal{P}_5$ (keys) through a cross attention operation, which is formulated as:

$$\boldsymbol{p}_i' = \mathrm{CrossAttention}(\boldsymbol{p}_i, \mathcal{P}_5).$$

Note that in the original DETR, the object features are randomly initialized, learnable object queries.

HPR. Each refiner block includes a self-attention (SA) layer, a refiner (R), and an FFN layer. Let $\boldsymbol{f}$ denote the feature of an object, m the main refiner, a_1 and a_2 the auxiliary refiners.

$$\boldsymbol{f}^{a_1}=\mathrm{SA}^{a_1}(\boldsymbol{f}),\ \boldsymbol{f}^{a_2}=\mathrm{SA}^{a_2}(\boldsymbol{f}),$$

$$\boldsymbol{f}^{m}=\alpha_m\mathrm{SA}^{m}(\boldsymbol{f})+\alpha_1\boldsymbol{f}^{a_1}+\alpha_2\boldsymbol{f}^{a_2}$$

$$\boldsymbol{h}^{a_1}=\mathrm{FFN}^{a_1}(\mathrm{R}^{a_1}(\boldsymbol{f}^{a_1})),\ \boldsymbol{h}^{a_2}=\mathrm{FFN}^{a_2}(\mathrm{R}^{a_2}(\boldsymbol{f}^{a_2}))$$

$$\boldsymbol{h}^{m}=\beta_m\mathrm{FFN}^{m}(\mathrm{R}^{m}(\boldsymbol{f}^{m}))+\beta_1\boldsymbol{h}^{a_1}+\beta_2\boldsymbol{h}^{a_2}.$$

All α and β are learnable. $\boldsymbol{h}^m$ acts as $\boldsymbol{f}$ in the next block.

3.4 More Implementation Details

Data Augmentations. We summarize the normal (DETR-style), strong (used in our data re-augmentation), and large-scale jitter (LSJ) [14] data augmentations in Table 3.15. We apply the LSJ data augmentation to the image batch that has been processed with the proposed data re-augmentation.

Hyper-Parameters. All hyper-parameters used in our model are presented in Table 3.16.

TABLE 3.15. Summary of various data augmentations applied in our model. $*$: the use of a larger augmentation factor.

Normal Augmentation	Random Flip, Random Resize, Random Crop	
Strong Augmentation	Geometric	Random Erasing, Rotate, Shear X, Shear Y, Translate X, Translate Y
	Appearance	Color Transform, Auto Contrast, Equalize, Sharpness, Posterize, Solarize, Color Balance, Contrast, Brightness
	Random Erasing	
LSJ Augmentation	Random Resize*, Random Crop*, Random Flip, Pad, Copy-Paste	

TABLE 3.16. Summary of hyper-parameters.

Hyper-Patameter	Value
Backbone Features	(Res3, Res4, Res5)
Freeze Batchnorm	Truth
Neck Features	($\mathcal{P}_3$, $\mathcal{P}_4$, $\mathcal{P}_5$, $\mathcal{P}_6$)
Query Number	900
Loss Weight	2:1:1
Position Embedding Offset	-0.5
Position Embedding Temperature	10000
Encoder Number	6
Decoder Number	6
Embedding Dimension	256
Head Number	8
FFN Dimension	2048
HPR Integration Weight	Learnable
HPR Integration Type	Vector
HPR Integration Initialization	1:1:1
RoI Resolution	7×7
Dynamic Conv. Feature Dimension	64
Denoising Query Number	100
Classification Cost (Hungarian)	2.0
Bbox Cost (Hungarian)	5.0
GIoU Cost (Hungarian)	2.0
Classification Loss	Cross Entropy
Loss Weight (Classification)	1.0
Loss Weight (Bbox)	5.0
Loss Weight (GIoU)	2.0
gamma (Align DETR)	2.0
tau (Align DETR)	1.5
alpha (Align DETR)	0.25
Repeat GT Number (Align DETR)	2

CHAPTER 4

Conclusion and Future works

This thesis has presented a comprehensive investigation into the advancement of object detection techniques, with a primary focus on the development of the Hybrid Proposal Refiner (HPR). The proposed HPR integrates multiple refinement techniques to optimize proposal generation, achieving significant improvements in detection accuracy. Through its integration into DETR-based detectors, the HPR demonstrates a substantial performance enhancement, with an Average Precision (AP) of 54.9 on the COCO 2017 benchmark using a ResNet-50 backbone and 60.0 AP with a Swin-Large backbone. Furthermore, we introduced an innovative data re-augmentation strategy that further improved the model's robustness and generalization capabilities, emphasizing the practical benefits of the proposed methods in real-world detection tasks.

In addition to the technical contributions, this thesis also offers valuable insights into the design and evolution of object detection systems. By analyzing the progression from CNN-based architectures to Transformer-based frameworks, particularly the transition from Faster R-CNN to the DETR series, we have highlighted the critical role of proposal refinement and matching algorithms in determining detection performance. The findings demonstrate that a well-constructed synergy between various proposal refiners is essential for enhancing the robustness and accuracy of detection models. These results not only extend the understanding of the inner workings of DETR-based models but also provide a solid foundation for future developments in the field.

Looking forward, several promising directions for future research emerge from this work. First, further refinement of the HPR, particularly through the integration of additional proposal

refinement techniques or more advanced matching algorithms, could yield even higher performance levels, particularly in complex detection scenarios. Additionally, the development of new, more challenging datasets that better reflect real-world conditions—such as those incorporating diverse object scales, occlusions, or dynamic environments—would provide a more rigorous testing ground for the next generation of detection models. Finally, as the use of Transformer-based architectures continues to expand in the field of computer vision, there is significant potential for extending the HPR framework to tasks beyond static image detection, such as video-based detection, real-time tracking, and autonomous navigation. These research directions offer the opportunity to push the boundaries of object detection systems and contribute to the ongoing advancement of intelligent, adaptive AI systems capable of performing in complex, real-world environments.

Bibliography

[1] Glenn Jocher et. al. *ultralytics/yolov5: v6.0 - YOLOv5n 'Nano' models, Roboflow integration, TensorFlow export, OpenCV DNN support*. Version v6.0. Oct. 2021. DOI: 10.5281/zenodo.5563715. URL: https://doi.org/10.5281/zenodo.5563715.

[2] Alexey Bochkovskiy, Chien-Yao Wang and Hong-Yuan Mark Liao. 'Yolov4: Optimal speed and accuracy of object detection'. In: *arXiv preprint arXiv:2004.10934* (2020).

[3] Zhaowei Cai and Nuno Vasconcelos. 'Cascade r-cnn: Delving into high quality object detection'. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 6154–6162.

[4] Zhi Cai et al. 'Align-DETR: Improving DETR with Simple IoU-aware BCE loss'. In: *arXiv preprint arXiv:2304.07527* (2023).

[5] Nicolas Carion et al. 'End-to-end object detection with transformers'. In: *European Conference on Computer Vision*. 2020, pp. 213–229.

[6] Chun-Fu Richard Chen, Quanfu Fan and Rameswar Panda. 'Crossvit: Cross-attention multi-scale vision transformer for image classification'. In: *IEEE International Conference on Computer Vision*. 2021, pp. 357–366.

[7] Kai Chen et al. 'Hybrid task cascade for instance segmentation'. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4974–4983.

[8] Qiang Chen et al. 'Group detr: Fast detr training with group-wise one-to-many assignment'. In: *IEEE International Conference on Computer Vision*. 2023, pp. 6633–6642.

[9] Xiaokang Chen et al. 'Conditional detr v2: Efficient detection transformer with box queries'. In: *arXiv preprint arXiv:2207.08914* (2022).

[10] Stéphane d'Ascoli et al. 'Convit: Improving vision transformers with soft convolutional inductive biases'. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 2286–2296.

[11] Jia Deng et al. 'Imagenet: A large-scale hierarchical image database'. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255.

[12] Alexey Dosovitskiy et al. 'An image is worth 16x16 words: Transformers for image recognition at scale'. In: *arXiv preprint arXiv:2010.11929* (2020).

[13] Yu Du et al. 'Learning to prompt for open-vocabulary object detection with vision-language model'. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2022, pp. 14084–14093.

[14] Golnaz Ghiasi et al. 'Simple copy-paste is a strong data augmentation method for instance segmentation'. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2021, pp. 2918–2928.

[15] Ross Girshick. 'Fast r-cnn'. In: *IEEE International Conference on Computer Vision*. 2015, pp. 1440–1448.

[16] Ross Girshick et al. 'Rich feature hierarchies for accurate object detection and semantic segmentation'. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 580–587.

[17] Kai Han et al. 'Transformer in transformer'. In: *Advances in Neural Information Processing Systems*. Vol. 34. 2021, pp. 15908–15919.

[18] Kaiming He et al. 'Deep residual learning for image recognition'. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.

[19] Kaiming He et al. 'Mask r-cnn'. In: *IEEE International Conference on Computer Vision*. 2017, pp. 2961–2969.

[20] Elad Hoffer et al. 'Augment your batch: Improving generalization through instance repetition'. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2020, pp. 8129–8138.

[21] Lichao Huang et al. 'Densebox: Unifying landmark localization with end to end object detection'. In: *arXiv preprint arXiv:1509.04874* (2015).

[22] Ding Jia et al. ‘Detrs with hybrid matching’. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2023, pp. 19702–19712.

[23] Tao Kong et al. ‘Foveabox: Beyound anchor-based object detection’. In: *IEEE Transactions on Image Processing* 29 (2020), pp. 7389–7398.

[24] Harold W Kuhn. ‘The Hungarian method for the assignment problem’. In: *Naval research logistics quarterly* 2.1-2 (1955), pp. 83–97.

[25] Hei Law and Jia Deng. ‘Cornernet: Detecting objects as paired keypoints’. In: *European Conference on Computer Vision*. 2018, pp. 734–750.

[26] Chuyi Li et al. ‘YOLOv6: A single-stage object detection framework for industrial applications’. In: *arXiv preprint arXiv:2209.02976* (2022).

[27] Feng Li et al. ‘Dn-detr: Accelerate detr training by introducing query denoising’. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2022, pp. 13619–13627.

[28] Feng Li et al. ‘Mask dino: Towards a unified transformer-based framework for object detection and segmentation’. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2023, pp. 3041–3050.

[29] Tsung-Yi Lin et al. ‘Feature pyramid networks for object detection’. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2117–2125.

[30] Tsung-Yi Lin et al. ‘Focal loss for dense object detection’. In: *IEEE International Conference on Computer Vision*. 2017, pp. 2980–2988.

[31] Tsung-Yi Lin et al. ‘Microsoft COCO: Common objects in context’. In: *European Conference on Computer Vision*. 2014, pp. 740–755.

[32] Yutong Lin et al. ‘DETR does not need multi-scale or locality design’. In: *IEEE International Conference on Computer Vision*. 2023, pp. 6545–6554.

[33] Shilong Liu et al. ‘Dab-detr: Dynamic anchor boxes are better queries for detr’. In: *International Conference on Learning Representations*. 2022, pp. 1–20.

[34] Wei Liu et al. ‘Ssd: Single shot multibox detector’. In: *European Conference on Computer Vision*. 2016, pp. 21–37.

[35] Ze Liu et al. ‘Swin transformer: Hierarchical vision transformer using shifted windows’. In: *IEEE International Conference on Computer Vision*. 2021, pp. 10012–10022.

[36] Ilya Loshchilov and Frank Hutter. ‘Decoupled weight decay regularization’. In: *arXiv preprint arXiv:1711.05101* (2017).

[37] Xin Lu et al. ‘Grid r-cnn’. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 7363–7372.

[38] Depu Meng et al. ‘Conditional detr for fast training convergence’. In: *IEEE International Conference on Computer Vision*. 2021, pp. 3651–3660.

[39] Jeffrey Ouyang-Zhang et al. ‘NMS Strikes Back’. In: *arXiv preprint arXiv:2212.06137* (2022).

[40] Joseph Redmon and Ali Farhadi. ‘YOLO9000: better, faster, stronger’. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 7263–7271.

[41] Joseph Redmon and Ali Farhadi. ‘Yolov3: An incremental improvement’. In: *arXiv preprint arXiv:1804.02767* (2018).

[42] Joseph Redmon et al. ‘You only look once: Unified, real-time object detection’. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 779–788.

[43] Shaoqing Ren et al. ‘Faster r-cnn: Towards real-time object detection with region proposal networks’. In: *Advances in Neural Information Processing Systems* (2015), pp. 91–99.

[44] Debjyoti Sinha and Mohamed El-Sharkawy. ‘Thin mobilenet: An enhanced mobilenet architecture’. In: *IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference*. 2019, pp. 0280–0285.

[45] Peize Sun et al. ‘Sparse r-cnn: End-to-end object detection with learnable proposals’. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2021, pp. 14454–14463.

[46] Zhi Tian et al. ‘Fcos: Fully convolutional one-stage object detection’. In: *IEEE International Conference on Computer Vision*. 2019, pp. 9627–9636.

[47] Hugo Touvron et al. ‘Going deeper with image transformers’. In: *IEEE International Conference on Computer Vision*. 2021, pp. 32–42.

[48] Ashish Vaswani et al. ‘Attention is all you need’. In: *Advances in Neural Information Processing Systems*. 2017, pp. 5998–6008.

[49] Chien-Yao Wang, Alexey Bochkovskiy and Hong-Yuan Mark Liao. 'YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors'. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2023, pp. 7464–7475.

[50] Tao Wang et al. 'Pnp-detr: Towards efficient visual analysis with transformers'. In: *IEEE International Conference on Computer Vision*. 2021, pp. 4661–4670.

[51] Yingming Wang et al. 'Anchor detr: Query design for transformer-based detector'. In: *AAAI Conference on Artificial Intelligence*. 2022, pp. 2567–2575.

[52] Fangyun Wei et al. 'Aligning pretraining for detection via object-level contrastive learning'. In: *Advances in Neural Information Processing Systems*. 2021, pp. 22682–22694.

[53] Fangyun Wei et al. 'Point-set anchors for object detection, instance segmentation and pose estimation'. In: *European Conference on Computer Vision*. 2020, pp. 527–544.

[54] Xingyi Yang, Jingwen Ye and Xinchao Wang. 'Factorizing knowledge in neural networks'. In: *European Conference on Computer Vision*. 2022, pp. 73–91.

[55] Zhuyu Yao et al. 'Efficient detr: improving end-to-end object detector with dense prior'. In: *arXiv preprint arXiv:2104.01318* (2021).

[56] Hao Zhang et al. 'DINO: Detr with improved denoising anchor boxes for end-to-end object detection'. In: *arXiv preprint arXiv:2203.03605* (2022).

[57] Hongkai Zhang et al. 'Dynamic R-CNN: Towards high quality object detection via dynamic training'. In: *European Conference on Computer Vision*. 2020, pp. 260–275.

[58] Shifeng Zhang et al. 'Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection'. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2020, pp. 9759–9768.

[59] Shilong Zhang et al. 'Dense Distinct Query for End-to-End Object Detection'. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2023, pp. 7329–7338.

[60] Xingyi Zhou, Dequan Wang and Philipp Krähenbühl. 'Objects as points'. In: *arXiv preprint arXiv:1904.07850* (2019).

[61] Xizhou Zhu et al. 'Deformable DETR: Deformable Transformers for End-to-End Object Detection'. In: *International Conference on Learning Representations*. 2020, pp. 1–16.

[62] Zhuofan Zong, Guanglu Song and Yu Liu. ‘Detrs with collaborative hybrid assignments training’. In: *IEEE International Conference on Computer Vision*. 2023, pp. 6748–6758.