RT-DETRv3: Real-time End-to-End Object Detection with Hierarchical Dense Positive Supervision

Shuo Wang* Chunlong Xia* Feng Lv Yifeng Shi†
Baidu Inc, China
{wangshuo36, xiachunlong, lvfeng02, shiyifeng}@baidu.com

Abstract

RT-DETR is the first real-time end-to-end transformer-based object detector. Its efficiency comes from the framework design and the Hungarian matching. However, compared to dense supervision detectors like the YOLO series, the Hungarian matching provides much sparser supervision, leading to insufficient model training and difficult to achieve optimal results. To address these issues, we proposed a hierarchical dense positive supervision method based on RT-DETR, named RT-DETRv3. Firstly, we introduce a CNN-based auxiliary branch that provides dense supervision that collaborates with the original decoder to enhance the encoder's feature representation. Secondly, to address insufficient decoder training, we propose a novel learning strategy involving self-attention perturbation. This strategy diversifies label assignment for positive samples across multiple query groups, thereby enriching positive supervisions. Additionally, we introduce a shared-weight decoder branch for dense positive supervision to ensure more high-quality queries matching each ground truth. Notably, all aforementioned modules are training-only. We conduct extensive experiments to demonstrate the effectiveness of our approach on COCO val2017. RT-DETRv3 significantly outperforms existing real-time detectors, including the RT-DETR series and the YOLO series. For example, RT-DETRv3-R18 achieves 48.1% AP (+1.6%/+1.4%) compared to RT-DETR-R18/RT-DETRv2-R18, while maintaining the same latency. Furthermore, RT-DETRv3-R101 can attain an impressive 54.6% AP outperforming YOLOv10-X. The code will be released at https://github.com/clxia12/RT-DETRv3.

1. Introduction

Object detection is an important fundamental problem in computer vision, which mainly focuses on obtaining the position and category information of objects in the image. Real-time object detection has higher requirements for algorithm performance, such as inference speed greater than 30 FPS. It has enormous value in practical applications such as autonomous driving, video surveillance, and object tracking. In recent years, real-time object detections have garnered significant attention from both researchers and industry professionals due to its efficient inference speed and superior detection accuracy. Among these, the most popular are single-stage real-time object detectors based on CNNs, such as the YOLO series ([1, 11–13, 22, 24, 26]). They all adopted a one-to-many label assignment strategy, designed an efficient inference framework, and used non-maximum suppression (NMS) to filter redundant prediction results. Although this strategy introduced additional latency, they achieved a trade-off between accuracy and speed.

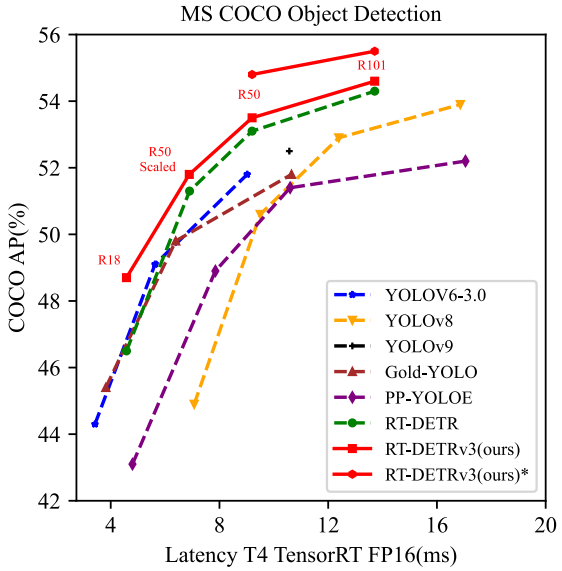

Figure 1. **Compared to other real-time object detectors.** Our method has better performance in the trade-off between speed and accuracy. ∗ represents adding extra data.

*Equal Contribution.
†Corresponding Author.

RT-DETRv3：基于层次化密集正监督的实时端到端目标检测

王硕* 夏春龙* 吕峰 史一峰† 百度公司，中国

{哇 ngshuo36、xiachunlong、lvfeng02、shiyifeng}@bai 都网

摘要

RT-DETR is the first real-time end-to-end transformer-based object detector. Its efficiency comes from the framework design and the Hungarian matching. However, compared to dense supervision detectors like the YOLO series, the Hungarian matching provides much sparser supervision, leading to insufficient model training and difficult to achieve optimal results. To address these issues, we proposed a hierarchical dense positive supervision method based on RT-DETR, named RT-DETRv3. Firstly, we introduce a CNN-based auxiliary branch that provides dense supervision that collaborates with the original decoder to enhance the encoder's feature representation. Secondly, to address insufficient decoder training, we propose a novel learning strategy involving self-attention perturbation. This strategy diversifies label assignment for positive samples across multiple query groups, thereby enriching positive supervisions. Additionally, we introduce a shared-weight decoder branch for dense positive supervision to ensure more high-quality queries matching each ground truth. Notably, all aforementioned modules are training-only. We conduct extensive experiments to demonstrate the effectiveness of our approach on COCO val2017. RT-DETRv3 significantly outperforms existing real-time detectors, including the RT-DETR series and the YOLO series. For example, RT-DETRv3-R18 achieves 48.1% AP (+1.6%/+1.4%) compared to RT-DETR-R18/RT-DETRv2-R18, while maintaining the same latency. Furthermore, RT-DETRv3-R101 can attain an impressive 54.6% AP outperforming YOLOv10-X. The code will be released at https://github.com/clxia12/RT-DETRv3.

1. 引言

目标检测是计算机视觉中的一个重要基础问题，主要致力于获取{v*}

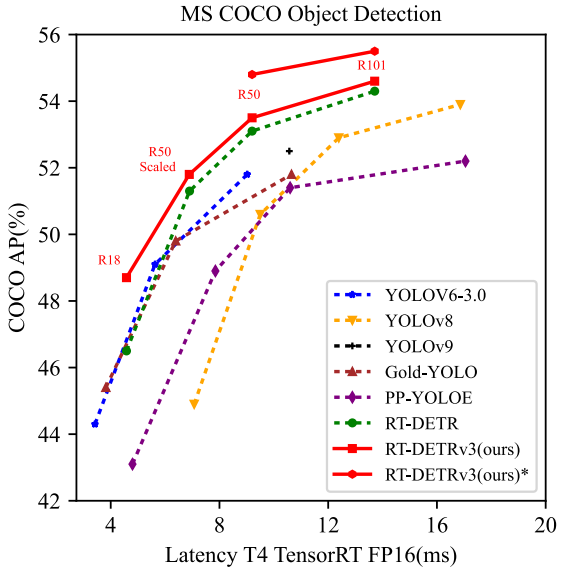

图1. 与其他实时目标检测器相比，我们的方法在速度与精度权衡上表现更优。∗表示额外添加数据的情况。

图像中物体的位置与类别信息。实时目标检测对算法性能有着更高要求，例如推理速度需超过30帧/秒。其在自动驾驶、视频监控、目标跟踪等实际应用中具有巨大价值。近年来，实时目标检测凭借高效的推理速度和卓越的检测精度，引起了研究人员与行业专业人士的高度关注。其中最为流行的是基于CNN的单阶段实时检测器，如YOLO系列（[1, 11–13, 22, 24, 26]）。它们均采用一对多的标签分配策略，设计了高效推理框架，并利用非极大值抑制（NMS）过滤冗余预测结果。尽管该策略会引入额外延迟，但实现了精度与速度的平衡。

*Equal Contribution.
†Corresponding Author.

DETR [2] is the first transformer-based end-to-end object detection algorithm. It employs set prediction and is optimized through the Hungarian matching strategy, eliminating the need for NMS post-processing and thereby simplifying the object detection process. Subsequent DETRs (such as DAB-DETR [16], DINO [29], and DN-DETR [14], etc.) further introduce iterative refinement schemes and denoising training, which effectively accelerating the convergence speed of the model and improving its performance. However, its high computational complexity significantly limits its practical applications.

RT-DETR [32] is the first real-time end-to-end transformer-based object detection algorithm. It designed an efficient hybrid encoder and IoU-aware query selection module, and a scalable decoder layer, achieving better results than other real-time detectors. However, the Hungarian matching strategy provides sparse supervision during training, leading to insufficient training of both the encoder and decoder, which limits the optimal performance of the approach. RT-DETRv2 [19] further enhances the flexibility and practicality of RT-DETR [32] by optimizing the training strategy to improve performance without sacrificing speed, although requires longer training time. To effectively address the issue of sparse supervision in object detection, we propose a hierarchical dense positive supervision method, which effectively accelerates model convergence and enhances model performance by introducing multiple auxiliary branches during training. Our main contributions are as follows:

- We introduce a one-to-many label assignment auxiliary head based on CNN, which collaborates with the original detection branch for optimization, further enhancing the representational capability of the encoder.
- We propose a learning strategy with self-attention perturbations aimed at enhancing the supervision of the decoder by diversifying label assignments across multiple query groups. Additionally, we introduced a shared-weight decoder branch for dense positive supervision to ensure more high-quality queries matching each ground truth. These approaches significantly improve the model's performance and accelerate convergence without additional inference latency.
- Extensive experiments conducted on the COCO dataset have thoroughly validated the effectiveness of our proposed approach. As shown in Figure 1, RT-DETRv3 significantly outperforms other real-time detectors, including the RT-DETR series and YOLO series. For instance, RT-DETRv3-R18 achieves 48.1% AP (+1.6%) compared to RT-DETR-R18, while maintaining the same latency. Additionally, RT-DETRv3-R50 outperforms YOLOv9-C by 0.9% AP, even with a latency reduction of 1.3ms.

2. Related Work

2.1. CNN-based real-time object detection.

The current CNN-based real-time object detectors are mainly the YOLO series. YOLOv4 [1] and YOLOv5 [12] optimized the network architecture (e.g., by adopting CSPNet [25] and PAN [17]), while also utilizing Mosaic data augmentation. YOLOv6 [13] further optimized the structure, including the RepVGG [6] backbone, decoupled head, SimSPPF, and more effective training strategy (e.g., SimOTA [7], etc.). YOLOv7 [24] introduces the E-ELAN attention module to better integrate features from different levels and adopts the adaptive anchor mechanism to improve small object detection. YOLOv8 [11] proposed a C2f module for effective feature extraction and fusion. YOLOv9 [26] proposed a new GELAN architecture and designed a PGI to enhance the training process. The PP-YOLO series [10,18] is a real-time object detection solution based on the PaddlePaddle framework proposed by Baidu. This series of algorithms has been optimized and improved on the basis of the YOLO series, aiming to improve detection accuracy and speed to meet the needs of practical application scenarios.

2.2. Transformer-based real-time object detection.

RT-DETR [32] is the first real-time end-to-end object detector. This approach designs an efficient hybrid encoder that effectively processes multi-scale features by decoupling intra-scale interactions and cross-scale fusion and proposes IoU-aware query selection to further improve performance by providing higher-quality initial object queries to the decoder. Its accuracy and speed are superior to the YOLO series of the same period, and it has received widespread attention. RT-DETRv2 [19] further optimized the training strategy, including dynamic data augmentation and optimized sampling operators for easy deployment, resulting in further improvement of its model performance. However, due to their one-to-one sparse supervision, the convergence speed and final effect are limited. Therefore, introducing a one-to-many label assignment strategy can further improve the model's performance.

2.3. Auxiliary training strategy.

Co-DETR [33] proposed multiple parallel one-to-many label assignment auxiliary head training strategies (e.g., ATSS [30] and Faster RCNN [20]), which can easily enhance the learning ability of the encoder in end-to-end detectors. For example, the integration of ViT-CoMer [27] with Co-DETR [33] has achieved state-of-the-art performance on the COCO detection task. DAC-DETR [9], MS-DETR [31], and GroupDETR [4] mainly accelerate the convergence of the model by adding one-to-many supervised information to the decoder of the model. The above ap-

DETR [2]是首个基于Transformer的端到端目标检测算法。它采用集合预测方式，并通过匈牙利匹配策略进行优化，无需NMS后处理步骤，从而简化了目标检测流程。后续的DETR变体（如DAB-DETR [16]、DINO [29]和DN-DETR [14]等）进一步引入了迭代优化方案和去噪训练机制，有效提升了模型收敛速度与检测性能。然而，其高昂的计算复杂度极大限制了实际应用场景。

RT-DETR[32]是首个基于Transformer的实时端到端目标检测算法。它设计了高效的混合编码器、IoU感知查询选择模块和可扩展的解码器层，取得了优于其他实时检测器的效果。然而，匈牙利匹配策略在训练过程中提供的监督信号较为稀疏，导致编码器和解码器训练不足，限制了该方法的性能上限。RT-DETRv2[19]通过优化训练策略，在不牺牲速度的前提下提升性能，进一步增强了RT-DETR[32]的灵活性和实用性，但需要更长的训练时间。为解决目标检测中监督稀疏性问题，我们提出了一种层次化密集正监督方法，通过在训练时引入多个辅助分支，有效加速模型收敛并提升性能。主要贡献如下:

- 我们引入了一个基于CNN的一对多标签分配辅助头，它与原始检测分支协同优化，进一步增强了编码器的表征能力。
- 我们提出了一种带有自注意力扰动的学习策略，旨在通过多样化多个查询组间的标签分配来增强解码器的监督效果。此外，我们引入了一个共享权重的解码器分支，用于密集正样本监督，以确保更多高质量查询与每个真实标注匹配。这些方法显著提升了模型性能，并在不增加推理延迟的情况下加速了收敛过程。
- 在COCO数据集上进行的大量实验充分验证了我们所提方法的有效性。如图1所示，RT-DETRv3显著优于其他实时检测器，包括RT-DETR系列和YOLO系列。例如，RT-DETRv3-R18在保持相同延迟的同时，AP达到48.1%（+1.6%），优于RT-DETR-R18。此外，RT-DETRv3-R50的AP比YOLOv9-C高出0.9%，同时延迟还降低了1.3毫秒。

2. 相关工作

2.1. 基于CNN的实时目标检测。

当前基于CNN的实时目标检测器主要为YOLO系列。YOLOv4[1]和YOLOv5[12]优化了网络架构（如采用CSPNet[25]和PAN[17]），同时运用了Mosaic数据增强技术。YOLOv6[13]进一步优化了结构，包括RepVGG[6]骨干网络、解耦头、SimSPPF模块，以及更高效的训练策略（如SimOTA[7]等）。YOLOv7[24]引入E-ELAN注意力模块以更好地融合多层级特征，并采用自适应锚框机制提升小目标检测性能。YOLOv8[11]提出C2f模块实现高效特征提取与融合。YOLOv9[26]创新性地提出GELAN架构，并设计PGI机制以增强训练过程。PP-YOLO系列[10,18]是百度基于PaddlePaddle框架提出的实时目标检测方案，该系列算法在YOLO基础上进行优化改进，旨在提升检测精度与速度，满足实际应用场景需求。

2.2. 基于Transformer的实时目标检测

RT-DETR[32]是首个实时端到端目标检测器。该方法设计了一种高效的混合编码器，通过解耦尺度内交互与跨尺度融合，有效处理多尺度特征，并提出IoU感知查询选择机制，通过为解码器提供更高质量的初始目标查询进一步提升性能。其精度与速度均优于同期YOLO系列，获得了广泛关注。RT-DETRv2[19]进一步优化了训练策略，包括动态数据增强和便于部署的优化采样算子，使模型性能得到更进一步提升。但由于其采用一对一稀疏监督机制，收敛速度与最终效果受到限制。因此，引入一对多标签分配策略可进一步提升模型性能。

2.3. 辅助训练策略

Co-DETR [33]提出了多种并行的一对多标签分配辅助头训练策略（如ATSS [30]和Faster RCNN [20]），这些策略能够轻松增强端到端检测器中编码器的学习能力。例如，ViT-CoMer [27]与Co-DETR [33]的结合在COCO检测任务上实现了最先进的性能。DAC-DETR [9]、MS-DETR [31]和GroupDETR [4]主要通过向模型的解码器添加一对多监督信息来加速模型的收敛。上述方-

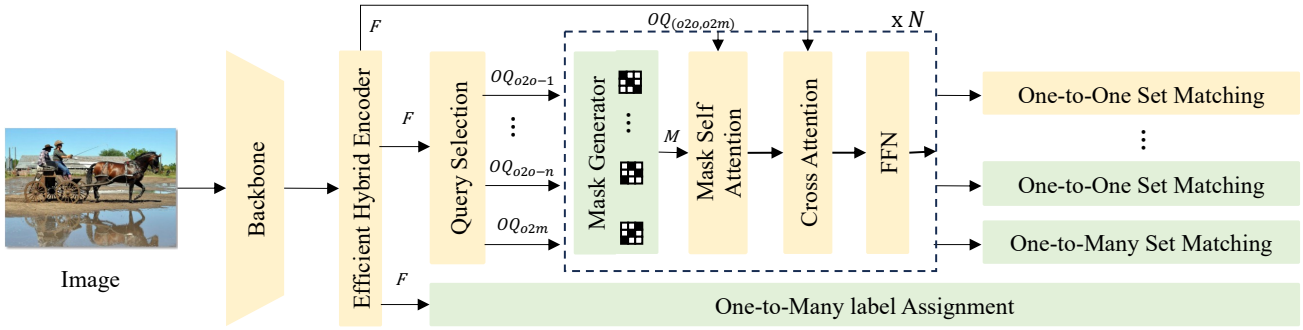

Figure 2. **Architecture of RT-DETRv3.** We preserve the core architecture of RT-DETR(highlighted in yellow)and propose a novel hierarchical decoupled dense supervision method (emphasized in green). Firstly, we enhance the encoder's representation capability by incorporating a CNN-based one-to-many label assignment auxiliary branch. Secondly, to enhance and strengthen supervision of the decoder, we generate multiple object queries (OQ) through the query selection module and apply random masking to perturb the self-attention mechanism, effectively diversifying the distribution of positive query samples. Additionally, to ensure that multiple relevant queries focus on the same target, we introduce a supplementary one-to-many matching branch. Notably, these auxiliary branches are discarded during evaluation.

proaches accelerate the convergence or improve the performance of the model by adding additional auxiliary branches at different positions of the model, but they are not real-time object detectors. Inspired by these, we introduced multiple one-to-many auxiliary dense supervision modules to both the encoder and decoder of RT-DETR [32]. These modules enhance the convergence speed and improve the overall performance of the RT-DETR [32]. Since these modules are only involved during the training phase, they don't affect the inference latency of RT-DETR [32].

3. Method

3.1. Overall Architecture.

The overall structure of RT-DETRv3 is shown in Figure 2. We have retained the overall framework of RT-DETR [32] (highlighted in yellow) and additionally introduced our proposed hierarchical decoupling dense supervision method (highlighted in green). Initially, the input image is processed through a CNN backbone (e.g., ResNet [8]) and a feature fusion module, termed the efficient hybrid encoder, to obtain multi-scale features $\{C_3, C_4, \text{and } C_5\}$. These features are then fed into a CNN-based one-to-many auxiliary branch and a transformer-based decoder branch in parallel. For the CNN-based one-to-many auxiliary branch, we directly employ existing state-of-the-art dense supervision methods, such as PP-YOLOE [28], to collaboratively supervise the encoder's representation learning. In the transformer-based decoder branch, the multi-scale features are first flattened and concatenated. We then use a query selection module to select the top-k features from them to generate object queries. Within the decoder, we introduce a mask generator that produces multiple sets of random masks. These masks are applied to the self-attention module, affecting the correlation between queries and thus differentiating the assignments of positive queries. Each set of random masks is paired with a corresponding query, as depicted in the Figure 2 by $OQ_{o2o-1}, ..., OQ_{o2o-n}$. Furthermore, to ensure that there are more high-quality queries matching each ground truth, we incorporate an one-to-many label assignment branch within the decoder. The following sections provide a detailed description of the modules proposed in this work.

3.2. Overview of RT-DETR.

RT-DETR [32] is a real-time detection framework designed for object detection tasks. It integrates the advantages of end-to-end prediction from DETR [3] while optimizing inference speed and detection accuracy. To achieve real-time performance, the encoder module is replaced with a lightweight CNN backbone, and an Efficient Hybrid Encoder module which designed for efficient feature fusion. RT-DETR [32] proposed an Uncertainty-minimal query selection module to select high-confidence feature as object queries, reducing the difficulty of query optimization. Subsequently, multiple layers of the decoder enhance these queries through self-attention, cross-attention and feed-forward network (FFN) modules, with the prediction results produced by MLP layers. During the training optimization process, RT-DETR [32] employs Hungarian matching for one-to-one assignment. For loss calculation, it uses L1 loss and GIoU loss to supervise box regression, and variable focus loss (VFL) to supervise the learning of the classification task.

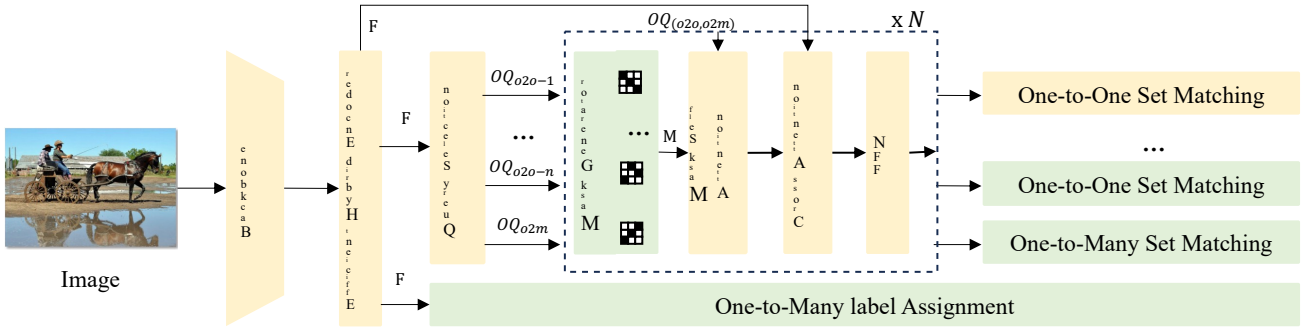

图2. RT-DETRv3架构图。我们保留了RT-DETR的核心架构（黄色高亮部分），并提出了一种新颖的分层解耦密集监督方法（绿色强调部分）。首先，通过引入基于CNN的一对多标签分配辅助分支，增强了编码器的表征能力。其次，为加强解码器的监督力度，我们通过查询选择模块生成多组对象查询(OQ)，并采用随机掩码干扰自注意力机制，有效丰富了正查询样本的分布多样性。此外，为确保多个相关查询聚焦同一目标，我们额外引入了一对多匹配分支作为补充。值得注意的是，这些辅助分支在评估阶段会被移除。

这些方法通过在模型的不同位置添加额外的辅助分支来加速收敛或提升模型性能，但它们并非实时目标检测器。受此启发，我们在RT-DETR[32]的编码器和解码器中引入了多重一对多辅助密集监督模块。这些模块不仅加快了RT-DETR[32]的收敛速度，还提升了其整体性能。由于这些模块仅在训练阶段参与，因此不会影响RT-DETR[32]的推理延迟。

域掩码。这些掩码被应用于自注意力模块，影响查询之间的相关性，从而区分正查询的分配。每组随机掩码都与一个对应的查询配对，如图2中的$OQ_{o2o-1}, ..., OQ_{o2o-n}$所示。此外，为了确保每个真实标注能匹配到更多高质量查询，我们在解码器中引入了一对多标签分配分支。接下来的章节将详细描述本工作中提出的各个模块。

3. 方法

3.1. 总体架构

RT-DETRv3的整体结构如图2所示。我们保留了RT-DETR[32]的整体框架（黄色高亮部分），并额外引入了提出的分层解耦密集监督方法（绿色高亮部分）。首先，输入图像通过CNN骨干网络（如ResNet[8]）和特征融合模块——即高效混合编码器进行处理，获得多尺度特征{C_3、C_4和C_5}。这些特征随后并行输入基于CNN的一对多辅助分支和基于Transformer的解码器分支。对于基于CNN的一对多辅助分支，我们直接采用现有最先进的密集监督方法（如PP-YOLOE[28]）来协同监督编码器的表征学习。在基于Transformer的解码器分支中，多尺度特征首先被展平并拼接。接着通过查询选择模块从中筛选出top-k特征以生成对象查询。在解码器内部，我们引入了能生成多组随机掩码的掩码生成器——

3.2. RT-DETR概述

RT-DETR [32] 是一种专为物体检测任务设计的实时检测框架。它融合了DETR [3]端到端预测的优势，同时优化了推理速度和检测精度。为实现实时性能，该框架采用轻量级CNN主干网络替换原编码器模块，并设计了高效混合编码器模块以实现特征的高效融合。RT-DETR [32]提出不确定性最小化查询选择模块，通过筛选高置信度特征作为物体查询，降低查询优化难度。随后通过多层解码器中的自注意力、交叉注意力和前馈网络(FFN)模块增强这些查询，最终由MLP层输出预测结果。在训练优化过程中，RT-DETR [32]采用匈牙利匹配算法进行一对一分配，并联合使用L1损失和GIoU损失监督边界框回归，同时采用可变焦点损失(VFL)监督分类任务的学习。

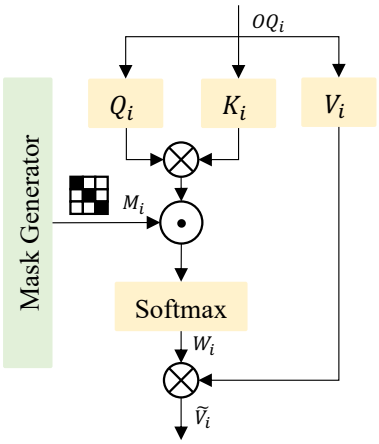

Figure 3. **Mask self-attention module.** M_i represents the perturbation mask corresponding to the i-th set of object queries. $\otimes$ denotes matrix multiplication, and $\odot$ denotes element-wise multiplication.

3.3. One-to-Many Auxiliary Branch Based on CNN.

To alleviate the problem of sparse supervision in encoder output caused by the decoder's one-to-one set matching scheme, we introduce an auxiliary detection head with one-to-many assignment, such as PP-YOLOE [28]. This strategy can effectively strengthen the supervision of the encoder, enabling it to have sufficient representation ability to accelerate the convergence of the model. Specifically, we directly integrate the output features $\{C_3, C_4, \text{and } C_5\}$ of the encoder into PP-YOLOE head, For the one-to-many matching algorithm, we follow the configuration of the PP-YOLOE head and use the ATSS matching algorithm in the early stage of training, and then switch to the TaskAlign matching algorithm. For the learning of classification and localization tasks, VFL and distributed focus loss (DFL) were respectively selected. Among them, VFL uses IoU scores as the target for positive samples, which makes positive samples with high IoU contribute relatively more to the loss. This also makes the model focus more on high-quality samples rather than low-quality samples during training. Specifically, decoder head also use VFL loss to ensure consistency in task definition. We denote the overall loss of the CNN auxiliary branch as L_{aux}, with the corresponding loss weight denoted as α.

3.4. Multi-Group Self-Attention Perturbation Branches Based on Transformer.

The decoder consists of a series of transformer blocks, with each block incorporating a self-attention, cross-attention, and FFN (Feed-Forward Network) module. Initially, the queries interact with each other through the self-attention module to enhance or diminish their feature representations. Subsequently, each query updates itself by retrieving information from the encoder's output features via the cross-attention module. Lastly, the FFN predicts the class and bounding box coordinates of the target corresponding to each query. However, the adoption of a one-to-one set matching in the RT-DETR leads to sparse supervision information, ultimately impairing the model's performance.

To ensure that multiple related queries associated with the same target have the opportunity to participate in positive sample learning, we propose multiple self-attention perturbation modules based on Mask Self-Attention. The implementation details of this perturbation module are shown in Figure 2. First, we generate multiple sets of object queries through the query selection module, denoted as OQ_i (i=1...N, where N is the number of sets). Correspondingly, we use a mask generator to generate a random perturbation mask M_i for each set of OQ_i. Both OQi and M_i are fed into the Mask Self-Attention module, resulting in the perturbed and fused features.

The detailed implementation of the Mask Self-Attention module is shown in Figure 3, OQ_i is first linearly projected to obtain Q_i, K_i, and V_i. Then, Q_i and K_i are multiplied to compute the attention weight, which is further multiplied by M_i and passed through a softmax function to yield the perturbed attention weight. Finally, this perturbed attention weight is multiplied by V_i to obtain the fused result $\tilde{V}_i$. The process can be represented as:

$$Q_i, K_i, V_i = Linear(OQ_i) \tag{1}$$

$$W_i = Softmax(M_i(Q_i K_i^T)) \tag{2}$$

$$\tilde{V}_i = W_i V_i \tag{3}$$

The introduction of multiple sets of random perturbations diversifies the features of the queries, allowing multiple related queries associated with the same target to have a chance of being assigned as positive sample queries, thereby enriching the supervision information. During training, multiple sets of object queries are concatenated and fed into a single decoder branch, enabling parameter sharing and enhancing training efficiency. The loss computation and label assignment scheme remain consistent with RT-DETR. We denote the loss of the i-th set as $Loss^i_{o2o}$, and the total loss for N perturbation sets is calculated as follow:

$$L_{o2o} = \frac{1}{N} \sum_{1}^{N} L^i_{o2o} \tag{4}$$

with the corresponding loss weight denoted as β.

3.5. One-to-Many Dense Supervision Branch Based on Transformer.

To maximize benefits of multi-group self-attention perturbation branches, we introduce an additional dense su-

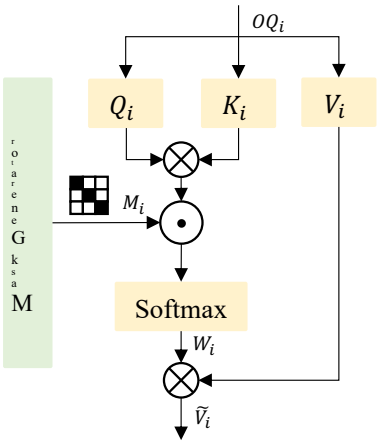

图3. 掩码自注意力模块。M_i代表与i-th组对象查询相对应的扰动掩码。⊗表示矩阵乘法，⊙表示逐元素乘法。

3.3. 基于CNN的一对多辅助分支

为缓解解码器一对一集合匹配方案导致编码器输出监督稀疏的问题，我们引入了具有一对多分配策略的辅助检测头，例如PP-YOLOE[28]。该策略能有效增强对编码器的监督，使其具备足够的表征能力以加速模型收敛。具体而言，我们将编码器的输出特征{C_3、C_4和C_5}直接接入PP-YOLOE检测头。针对一对多匹配算法，我们遵循PP-YOLOE头的配置方案：在训练初期采用ATSS匹配算法，后期切换至TaskAlign匹配算法。对于分类与定位任务的学习，分别选用VFL（Varifocal Loss）和分布式焦点损失（DFL）。其中VFL以IoU分数作为正样本目标，使得高IoU正样本对损失贡献更大，这促使模型在训练过程中更关注高质量样本而非低质量样本。值得注意的是，解码器头部同样采用VFL损失以确保任务定义的一致性。我们将CNN辅助分支的整体损失记为L_{aux}，其对应损失权重记为α。

3.4. 基于Transformer的多组自注意力扰动分支

解码器由一系列Transformer块组成，每个块包含自注意力模块、交叉注意力模块以及FFN（前馈网络）模块。最初，查询通过自注意力模块相互交互，以增强或削弱其特征表达——随后，每个查询通过交叉注意力模块从编码器的输出特征中检索信息来更新自身。最后，前馈网络（FFN）预测与每个查询相对应的目标类别和边界框坐标。然而，RT-DETR采用的一对一集合匹配机制导致了监督信息的稀疏性，最终影响了模型的性能。

为确保与同一目标相关的多个查询有机会参与正样本学习，我们提出了基于掩码自注意力（Mask Self-Attention）的多重自注意力扰动模块。该扰动模块的实现细节如图2所示。首先，通过查询选择模块生成多组对象查询，记为OQ_i (i=1...N，其中N为组数)。相应地，我们使用掩码生成器为每组OQ_i生成随机扰动掩码M_i。OQi与M_i共同输入掩码自注意力模块，最终得到扰动融合后的特征。

掩码自注意力模块的具体实现如图3所示，首先对OQ_i进行线性投影得到Q_i、K_i和V_i。接着，将Q_i与K_i相乘计算注意力权重，该权重再与M_i相乘并通过softmax函数处理，生成扰动后的注意力权重。最后，将此扰动权重与V_i相乘，得到融合结果$\tilde{V}_i$。该过程可表示为：

$$Q_i, K_i, V_i = Linear(OQ_i) \tag{1}$$

$$W_i = Softmax(M_i(Q_i K_i^T)) \tag{2}$$

$$\tilde{V}_i = W_i V_i \tag{3}$$

引入多组随机扰动使查询特征多样化，使得与同一目标相关联的多个相关查询有机会被分配为正样本查询，从而丰富了监督信息。训练过程中，多组对象查询被拼接后输入单一解码器分支，实现参数共享并提升训练效率。损失计算与标签分配方案保持与RT-DETR一致。我们将i-th集合的损失记为$Loss^i_{o2o}$，N组扰动集合的总损失计算如下：

$$L_{o2o} = \frac{1}{N}\sum_{1}^{N} L^i_{o2o} \tag{4}$$

对应的损失权重记为β。

3.5. 基于Transformer的一对多密集监督分支

为了最大化多组自注意力扰动分支的效益，我们引入了一个额外的密集

Model	Backbone	#Params(M)	GFLOPs	Latency (ms)	AP^{val} 1x	AP^{val} 3x	AP^{val} 6x
RT-DETR [32]	R18	20	60	4.6	38.7	44.5	46.5
RT-DETRv2 [19]					39.8	44.9	46.7/47.9†
RT-DETRv3 (ours)					**41.5**	**46.1**	**48.1/48.7**†
RT-DETR [32]	R34	31	92	6.3	42.8	47.5	48.9
RT-DETRv2 [19]					43.0	47.2	49.0/49.9†
RT-DETRv3 (ours)					**44.7**	**48.6**	**49.9/50.1**†
RT-DETR [32]	R50	42	136	9.2	48.9	52.2	53.1
RT-DETRv2 [19]					-	-	53.4
RT-DETRv3 (ours)					**50.2**	**53.0**	**53.4**
RT-DETR [32]	R101	76	259	13.5	50.2	53.5	54.3
RT-DETRv2 [19]					-	-	54.3
RT-DETRv3 (ours)					**51.3**	**54.2**	**54.6**

Table 1. **Comparison of the RT-DETR series for object detection on COCO val2017.** R18, R34, R50, and R101 refer to ResNet-18, ResNet-34, ResNet-50, and ResNet-101, respectively. 1x, 3x, and 6x respectively correspond to training for 12, 36, and 72 epochs. † denotes training for 120 epochs.

pervision branch with shared weights in the decoder. This ensures more high-quality queries matching each ground truth. Specifically, we employ a query selection module to generate a unique set of object queries. During the sample matching phase, an augmented target set is generated by replicating the training labels by a factor of m, with a default value of 4. This augmented set is subsequently matched against the prediction of the query. The loss computation remains consistent with the original detection loss, and we designate L_{o2m} as the loss function for this branch, with a loss weight of γ.

3.6. Total Loss.

In summary, the overall loss function of our proposed approach is as follows:

$$L = \alpha L_{aux} + \beta L_{o2o} + \gamma L_{o2m} \tag{5}$$

where L_{aux} is responsible for dense supervision of the encoder, L_{o2o} enriches the one-to-one supervision information for the decoder while preserving the end-to-end prediction characteristics, and L_{o2m} provides one-to-many dense supervision to the decoder. By default, the loss weights α, β, and γ are set to 1.

4. Experiments

4.1. Datasets and Evaluation Metrics.

We selected the MS COCO 2017 [15] object detection dataset as the evaluation benchmark for our approach. This dataset consists of 115k training images and 5k test images. We adopted the same evaluation metric, AP, as used in the RT-DETR [32] approach. We compared the performance of RT-DETRv3 with other real-time object detectors in terms of convergence efficiency, inference speed, and effectiveness, which include both transformer-based and CNN-based real-time object detectors. Additionally, we conducted ablation studies on the modules mentioned in this paper. All experimental details and results are elaborated in the following sections.

4.2. Implementation Details.

We integrated proposed hierarchical dense supervision branches into the RT-DETR [32] framework. The CNN-based dense supervision auxiliary branch directly employed the PP-YOLOE head, with its sample matching strategy, loss calculation, and all other configurations consistent with those of PP-YOLOE [28]. We reused the RT-DETR [32] decoder structure as the main branch and additionally added three groups of parameter-shared self-attention perturbation branches. The sample matching method is consistent with the main branch, utilizing Hungarian matching algorithm. We also added a parameter-shared one-to-many matching branch, where each ground truth is matched with four object queries by default and set 300 object queries in total. The AdamW optimizer, integrated with a weight decay factor of 0.0001, was employed, ensuring that all other training configurations adhered strictly to the RT-DETR [32], encompassing both data augmentation and pre-training. We used a 10x (120 epochs) training schedule for smaller backbones (R18, R34) and a 6x (72 epochs) training schedule for larger backbones (R50, R101). We use four NVIDIA

Model	Backbone	#Params(M)	GFLOPs	Latency (ms)	AP^{val} 1x	AP^{val} 3x	AP^{val} 6x
RT-DETR [32]	R18	20	60	4.6	38.7	44.5	46.5
RT-DETRv2 [19]					39.8	44.9	46.7/47.9†
RT-DETRv3 (ours)					**41.5**	**46.1**	**48.1/48.7**†
RT-DETR [32]	R34	31	92	6.3	42.8	47.5	48.9
RT-DETRv2 [19]					43.0	47.2	49.0/49.9†
RT-DETRv3 (ours)					**44.7**	**48.6**	**49.9/50.1**†
RT-DETR [32]	R50	42	136	9.2	48.9	52.2	53.1
RT-DETRv2 [19]					-	-	53.4
RT-DETRv3 (ours)					**50.2**	**53.0**	**53.4**
RT-DETR [32]	R101	76	259	13.5	50.2	53.5	54.3
RT-DETRv2 [19]					-	-	54.3
RT-DETRv3 (ours)					**51.3**	**54.2**	**54.6**

表1. RT-DETR系列在COCO val2017目标检测任务上的性能对比。R18、R34、R50和R101分别代表ResNet-18、ResNet-34、ResNet-50和ResNet-101网络架构。1x、3x和6x分别对应12、36和72轮次训练。†表示进行了120轮次训练。

解码器中采用共享权重的监督分支。这确保了更高质量的查询与每个真实标注相匹配。具体而言，我们利用查询选择模块生成一组独特的对象查询。在样本匹配阶段，通过将训练标签复制m倍（默认值为4）生成一个增强的目标集。随后，该增强集将与查询预测进行匹配。损失计算保持与原始检测损失一致，我们指定L_{o2m}作为该分支的损失函数，损失权重为γ。

3.6. 总损失

综上所述，我们提出的方法的总体损失函数如下：

$$L = \alpha L_{aux} + \beta L_{o2o} + \gamma L_{o2m} \tag{5}$$

其中L_{aux}负责对编码器进行密集监督，L_{o2o}在保留端到端预测特性的同时，为解码器丰富了一对一监督信息，而L_{o2m}则为解码器提供了一对多的密集监督。默认情况下，损失权重α、β和γ均设为1。

4. 实验

4.1. 数据集与评估指标

我们选用MS COCO 2017 [15]目标检测数据集作为方法的评估基准。该数据集包含115k训练图像和5k测试图像。我们采用了与RT-DETR[32]方法相同的评估指标AP。在收敛效率、推理速度和有效性方面，我们将RT-DETRv3与其他实时目标检测器进行了比较，这些检测器包括基于Transformer和基于CNN的实时目标检测器。此外，我们还对本文提到的模块进行了消融实验。所有实验细节和结果将在以下部分详细阐述。

4.2. 实现细节

我们将提出的分层密集监督分支集成到了RT-DETR[32]框架中。基于CNN的密集监督辅助分支直接采用了PP-YOLOE头部结构，其样本匹配策略、损失计算及其他所有配置均与PP-YOLOE[28]保持一致。我们复用RT-DETR[32]的解码器结构作为主分支，并额外添加了三组参数共享的自注意力扰动分支。样本匹配方式与主分支一致，采用匈牙利匹配算法。同时新增了一个参数共享的一对多匹配分支，默认每个真实标注框匹配四个目标查询，共设置300个目标查询。优化器采用集成权重衰减因子0.0001的AdamW，确保其他所有训练配置——包括数据增强与预训练——严格遵循RT-DETR[32]方案。较小骨干网络（R18、R34）采用10倍（120轮）训练周期，较大骨干网络（R50、R101）采用6倍（72轮）训练周期。我们使用四块NVIDIA

Model	#Epochs	#Params (M)	GFLOPs	Latency (ms)	AP^{val}
YOLOv6-3.0-S [13]	300	18.5	45.3	3.4	44.3
Gold-YOLO-S [23]	300	21.5	46.0	3.8	45.4
YOLO-MS-S [5]	300	8.1	31.2	10.1	46.2
YOLOv8-S [11]	500	11.2	28.6	7.1	46.2
YOLOv9-S [26]	500	7.1	26.4	-	46.7
YOLOV10-S [22]	500	7.2	21.6	2.5	46.3
RT-DETRv3-R18 (ours)	120	20	60	4.6	**48.7**
YOLOv6-3.0-M [13]	300	34.9	85.8	5.6	49.1
Gold-YOLO-M [23]	300	41.3	87.5	6.4	49.8
YOLO-MS [5]	300	22.2	80.2	12.4	51.0
YOLOv8-M [11]	500	25.9	78.9	9.5	50.6
YOLOv9-M [26]	500	20.0	76.3	-	51.1
YOLOV10-M [22]	500	15.4	59.1	4.7	51.1
RT-DETRv3-R34 (ours)	120	31	92	6.3	50.1
RT-DETRv3-R50m (ours)	72	36	100	6.89	**51.7**
Gold-YOLO-L [23]	300	75.1	151.7	9.0	51.8
YOLOv5-X [12]	300	86	205	23.3	50.7
PPYOLOE-L [28]	300	52	110	10.6	51.4
YOLOv6-L [13]	300	59	150	10.1	52.8
YOLOv7-L	300	36	104	18.2	51.2
YOLOV8-L [11]	500	43	165	14.1	52.9
YOLOv9-C [26]	500	25.3	102.1	10.57	52.5
YOLOV10-L [22]	500	24.4	120.3	7.28	53.2
RT-DETRv3-R50 (ours)	72	42	136	9.2	**53.4**
YOLOv8-X [11]	500	68.2	257.8	16.9	53.9
YOLOv10-X [22]	500	29.5	160.4	10.7	54.4
RT-DETRv3-R101 (ours)	72	76	259	13.5	**54.6**

Table 2. **Compared to CNN-based real-time object detectors on COCO val2017.**

A100 GPUs to train our proposed method with a batch size of 64. Moreover, the latencies of all models are tested on T4 GPU with TensorRT FP16, following [32]. We have observed that, in comparison to most detectors employing longer training epochs, RT-DETRv3 only needs 72 epochs to achieve superior accuracy.

4.3. Comparison with Transformer-based Real-time Object Detectors.

Inference speed and algorithm performance. The real-time object detectors based on transformer architecture are primarily represented by the RT-DETR series. Table 1 presents the comparison results between our approach and the RT-DETR series. Our approach outperforms both RT-DETR [32] and RT-DETRv2 [19] across various backbone. Specifically, compared to RT-DETR [32], with the 6x training schedule, our approach demonstrates improvements of 1.6%, 1.0%, 0.3%, and 0.3% with the R18, R34, R50, and R101 backbones, respectively. In comparison to RT-DETRv2 [19], we evaluated the R18 and R34 backbones under 6x/10x training schedules, where our approach improvements of 1.4%/0.8% and 0.9%/0.2%, respectively. Moreover, since the auxiliary dense supervision branches we proposed are training-only, our approach maintains the same inference speed as both RT-DETR [32] and RT-DETRv2 [19].

Convergence speed. Our approach builds on the RT-DETR [32] framework by incorporating CNN-based and

Model	#Epochs	#Params (M)	GFLOPs	Latency (ms)	AP^{val}
YOLOv6-3.0-S [13]	300	18.5	45.3	3.4	44.3
Gold-YOLO-S [23]	300	21.5	46.0	3.8	45.4
YOLO-MS-S [5]	300	8.1	31.2	10.1	46.2
YOLOv8-S [11]	500	11.2	28.6	7.1	46.2
YOLOv9-S [26]	500	7.1	26.4	-	46.7
YOLOV10-S [22]	500	7.2	21.6	2.5	46.3
RT-DETRv3-R18 (ours)	120	20	60	4.6	**48.7**
YOLOv6-3.0-M [13]	300	34.9	85.8	5.6	49.1
Gold-YOLO-M [23]	300	41.3	87.5	6.4	49.8
YOLO-MS [5]	300	22.2	80.2	12.4	51.0
YOLOv8-M [11]	500	25.9	78.9	9.5	50.6
YOLOv9-M [26]	500	20.0	76.3	-	51.1
YOLOV10-M [22]	500	15.4	59.1	4.7	51.1
RT-DETRv3-R34 (ours)	120	31	92	6.3	50.1
RT-DETRv3-R50m (ours)	72	36	100	6.89	**51.7**
Gold-YOLO-L [23]	300	75.1	151.7	9.0	51.8
YOLOv5-X [12]	300	86	205	23.3	50.7
PPYOLOE-L [28]	300	52	110	10.6	51.4
YOLOv6-L [13]	300	59	150	10.1	52.8
YOLOv7-L	300	36	104	18.2	51.2
YOLOV8-L [11]	500	43	165	14.1	52.9
YOLOv9-C [26]	500	25.3	102.1	10.57	52.5
YOLOV10-L [22]	500	24.4	120.3	7.28	53.2
RT-DETRv3-R50 (ours)	72	42	136	9.2	**53.4**
YOLOv8-X [11]	500	68.2	257.8	16.9	53.9
YOLOv10-X [22]	500	29.5	160.4	10.7	54.4
RT-DETRv3-R101 (ours)	72	76	259	13.5	**54.6**

表2. 与基于CNN的实时目标检测器在COCO val2017上的对比。

使用A100 GPU以64的批量大小训练我们提出的方法。此外，所有模型的延迟均在T4 GPU上采用TensorRT FP 16进行测试，遵循[32]的方法。我们观察到，与大多数采用更长训练周期的检测器相比，RT-DETRv3仅需72个周期即可达到卓越的准确度。

4.3. 与基于Transformer的实时目标检测器对比

推理速度与算法性能。基于Transformer架构的实时目标检测器主要以RT-DETR系列为代表。表1展示了我们的方法与RT-DETR系列的对比结果。我们的方法在{v*}方面均优于RT-DETR [32]和RT-DETRv2 [19]在不同骨干网络上的表现。具体而言，与RT-DETR [32]相比，在6倍训练周期下，我们的方法在R18、R34、R50和R101骨干网络上分别实现了1.6%、1.0%、0.3%和0.3%的性能提升。相较于RT-DETRv2 [19]，我们在6倍/10倍训练周期下评估了R18和R34骨干网络，我们的方法分别带来了1.4%/0.8%和0.9%/0.2%的改进。此外，由于我们提出的辅助密集监督分支仅用于训练阶段，我们的方法在推理速度上与RT-DETR [32]和RT-DETRv2 [19]保持一致。

收敛速度。我们的方法基于RT-DETR[32]框架，通过融入基于CNN和

Method	Extra data	Epochs	AP^{val}
RT-DETRv3-R50	x	51	53.4
	x	72	52.9 (-0.5)
	✓	51	54.2
	✓	72	54.8 (+0.6)
RT-DETRv3-R101	x	51	54.6
	x	72	54.2 (-0.4)
	✓	51	54.7
	✓	72	55.4 (+0.7)

Table 3. **Analysis of overfitting.** Extra data represents the Object365 dataset [21].

transformer-based one-to-many dense supervision, which not only boosts model performance but also speeds up convergence. We have conducted extensive experiments to validate the effectiveness of our approach. Table 1 presents a comparative analysis of RT-DETRv3, RT-DETR [32], and RT-DETRv2 [19] across various training schedules. It clearly demonstrates that our method outperforms them in terms of convergence speed in any schedule and only needs half of the training epochs to achieve the comparable performance.

Analysis of overfitting. As illustrated in Figure 4, we noticed that as the model size increases, RT-DETRv3 tends to exhibit overfitting. We believe this may be due to a mismatch between the size of the training dataset and the model size. We conducted several experiments, as shown in Table 3, when we added additional training data, the performance of RT-DETRv3 continues to improve as the training epochs increase, and it performs better than the model without the additional data at the same epochs.

4.4. Comparison with CNN-Based Real-time Object Detectors.

Inference speed and algorithm performance. We compared the end-to-end speed and accuracy of RT-DETRv3 with current advanced CNN-based real-time object detection methods. We categorized the models into small, medium, and large scales based on their inference speed. Under similar inference performance conditions, we compared RT-DETRv3 with other state-of-the-art algorithms such as YOLOv6-3.0 [13], Gold-YOLO [23], YOLO-MS [5], YOLOv8 [11], YOLOv9 [26], and YOLOv10 [22]. As shown in Table 2, for small-scale models, the RT-DETRv3-R18 approach outperforms YOLOv6-3.0-S, Gold-YOLO-S, YOLO-MS-S, YOLOv8-S, YOLOv9-S, and YOLOv10-S by 4.4%, 3.3%, 2.5%, 2.5%, 2.0%, and 2.4%, respectively. For medium-scale models, RT-DETRv3 also demonstrates superior performance compared to YOLOv6-3.0-M, Gold-YOLO-M, YOLO-MS-M, YOLOv8-M, YOLOv9-M, and YOLOv10-M. For large-scale models, our approach consistently outperforms CNN-based real-time object detectors. For example, our RT-DETRv3-R101 can achieve 54.6 AP, which is better than YOLOv10-X. However, since we have not yet optimized the overall framework of the RT-DETRv3 detector for lightweight deployment, there is still room for further improving the inference efficiency of RT-DETRv3.

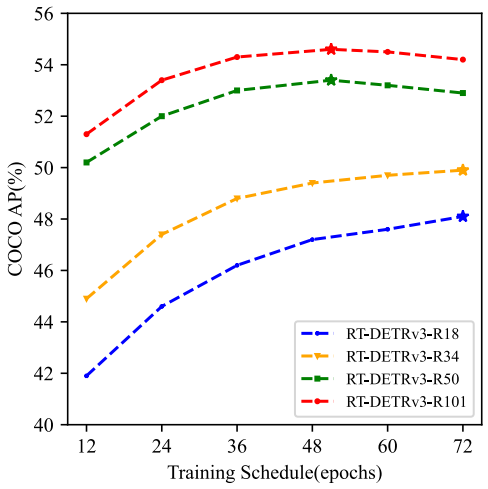

Figure 4. **Convergence curves of RT-DETRv3 across different model sizes.** ⋆ represents the best AP.

Convergence speed. As shown in Table 2, we are excited to find that our RT-DETRv3, while achieving superior performance, can reduce the training epochs to as little as 60% or even less compared to CNN-based real-time detectors.

4.5. Ablation Study.

Settings. We conducted the ablation experiments using RT-DETR [32] as the baseline and then validated the impact of our proposed approach by sequentially integrating auxiliary CNN-based one-to-many label assignments branch, the auxiliary transformer-based one-to-many label assignments branch, and the multi-group self-attention perturbation modules. These experiments were performed with ResNet18 as the backbone, with a batch size of 64, and four NVIDIA A100 GPUs, while maintaining other configurations consistent with RT-DETR [32].

Ablation for components. We conducted ablation experiments to evaluate proposed modules in this paper. As shown in Table 4, each module significantly enhances the model's performance. For instance, by adding O2M-T module to RT-DETR [32], we observed a 1.0% improvement in performance over the base model. When all proposed modules are integrated into RT-DETR for algorithm optimization, the model's performance improves by 1.6%.

Method	Extra data	Epochs	AP^{val}
RT-DETRv3-R50	x	51	53.4
	x	72	52.9 (-0.5)
	✓	51	54.2
	✓	72	54.8 (+0.6)
RT-DETRv3-R101	x	51	54.6
	x	72	54.2 (-0.4)
	✓	51	54.7
	✓	72	55.4 (+0.7)

表3. 过拟合分析。额外数据代表Object365数据集[21]。

基于transformer的一对多密集监督方法，不仅提升了模型性能，还加速了收敛速度。我们通过大量实验验证了该方法的有效性。表1展示了RT-DETRv3、RT-DETR[32]和RT-DETRv2[19]在不同训练周期下的对比分析，清晰表明我们的方法在任何训练周期下都具备更快的收敛速度，且仅需一半训练轮次即可达到相当性能水平。

过拟合分析。如图4所示，我们注意到随着模型规模增大，RT-DETRv3容易出现过拟合现象。我们认为这可能是训练数据集规模与模型规模不匹配所致。如表3所示，当我们增加额外训练数据时，RT-DETRv3的性能会随着训练轮次增加而持续提升，且在相同训练轮次下表现优于未添加额外数据的模型。

4.4. 与基于CNN的实时目标检测器比较

推理速度与算法性能。我们将RT-DETRv3的端到端速度和精度与当前先进的基于CNN的实时目标检测方法进行了比较。根据推理速度，将模型分为小、中、大规模。在相近的推理性能条件下，将RT-DETRv3与YOLOv6-3.0[13]、Gold-YOLO[23]、YOLO-MS[5]、YOLOv8[11]、YOLOv9[26]、YOLOv10[22]等前沿算法进行对比。如表2所示，在小规模模型中，RT-DETRv3-R18方法分别以4.4%、3.3%、2.5%、2.5%、2.0%和2.4%的优势超越YOLOv6-3.0-S、Gold-YOLO-S、YOLO-MS-S、YOLOv8-S、YOLOv9-S和YOLOv10-S。在中规模模型中，RT-DETRv3相较YOLOv6-3.0-M、Gold-YOLO-M同样展现出更优性能。

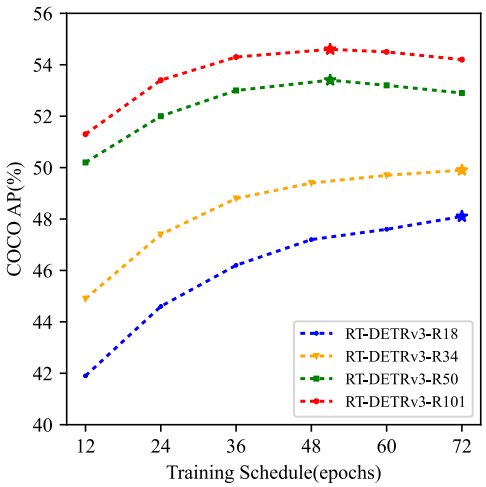

图4. RT-DETRv3在不同模型尺寸下的收敛曲线。⋆代表最佳平均精度(AP)。

YOLO-MS-M、YOLOv8-M、YOLOv9-M和YOLOv10-M。在大规模模型上，我们的方法持续超越基于CNN的实时目标检测器。例如，我们的RT-DETRv3-R101可实现54.6 AP，优于YOLOv10-X。但由于我们尚未针对轻量化部署优化RT-DETRv3检测器的整体框架，RT-DETRv3的推理效率仍有进一步提升空间。

收敛速度。如表2所示，我们欣喜地发现，在实现卓越性能的同时，我们的RT-DETRv3能将训练周期缩短至基于CNN的实时检测器的60%甚至更少。

4.5. 消融研究。

设置。我们以RT-DETR[32]为基线进行消融实验，依次整合基于辅助CNN的一对多标签分配分支、基于辅助Transformer的一对多标签分配分支以及多组自注意力扰动模块，验证所提方法的影响。实验采用ResNet18作为主干网络，批量大小为64，使用四块NVIDIA A100 GPU，其余配置与RT-DETR[32]保持一致。

组件消融实验。我们进行了消融实验以评估本文提出的各个模块。如表4所示，每个模块都显著提升了模型性能。例如，在RT-DETR[32]基础上添加O2M-T模块后，模型性能较基线提升了1.0%。当所有提出的模块被集成到RT-DETR中进行算法优化时，模型性能提升了1.6%。

Method	O2M-C	O2M-T	MGSA	AP^{val}
RT-DETR	x	x	x	46.5
	✓	x	x	47.4
	x	✓	x	47.5
	x	x	✓	47.5
	✓	✓	✓	**48.1**

Table 4. **Ablation studies of key components.** O2M-C represents the one-to-many auxiliary branch based on CNN, O2M-T refers to one-to-many dense supervision branch based on transformer, and MGSA stands for multi-group self-attention perturbation branch based on transformer.

Number	AP^{val}	AP^{val}_{50}
2	47.9	65.4
3	48.1	65.6
4	48.0	65.3

Table 5. **Ablation study on the number of self-attention perturbation branches.**

Number of self-attention perturbation branches. To verify the effect varying the number of self-attention perturbation branches on RT-DETRv3 performance, we conducted ablation experiments using RT-DETRv3-R18 with branch counts of 2, 3, and 4, while keeping all other configurations unchanged. As shown in Table 5, when the number of branches was set to 3, the model achieved its optimal performance with AP 48.1. Reducing the number of branches decreased the richness of the supervision signals, leading to lower performance. Conversely, increasing the number of branches excessively raised the model's learning difficulty without yielding significant performance gains.

5. Conclusion

In this paper, we propose a real-time object detection algorithm based on transformer, named RT-DETRv3. This algorithm builds upon RT-DETR by incorporating multiple dense positive sample auxiliary supervision modules. These modules apply one-to-many object supervision to specific features of both the encoder and decoder in RT-DETR, thereby accelerating the algorithm's convergence and improving its performance. It's important to note that these modules are training-only. We validated the effectiveness of our algorithm on the COCO object detection benchmark, and the experiments demonstrate that our algorithm achieves better results compared to other real-time object detectors. We hope that our work can inspire researchers and developers working on real-time transformer-based object detection.

References

[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 1, 2

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 2

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 3

[4] Qiang Chen, Xiaokang Chen, Jian Wang, Shan Zhang, Kun Yao, Haocheng Feng, Junyu Han, Errui Ding, Gang Zeng, and Jingdong Wang. Group detr: Fast detr training with group-wise one-to-many assignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6633–6642, 2023. 2

[5] Yuming Chen, Xinbin Yuan, Ruiqi Wu, Jiabao Wang, Qibin Hou, and Ming-Ming Cheng. Yolo-ms: rethinking multi-scale representation learning for real-time object detection. *arXiv preprint arXiv:2308.05480*, 2023. 6, 7

[6] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13733–13742, 2021. 2

[7] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 2

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[9] Zhengdong Hu, Yifan Sun, Jingdong Wang, and Yi Yang. Dac-detr: Divide the attention layers and conquer. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[10] Xin Huang, Xinxin Wang, Wenyu Lv, Xiaying Bai, Xiang Long, Kaipeng Deng, Qingqing Dang, Shumin Han, Qiwen Liu, Xiaoguang Hu, et al. Pp-yolov2: A practical object detector. *arXiv preprint arXiv:2104.10419*, 2021. 2

[11] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLO, Jan. 2023. 1, 2, 6, 7

[12] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, Yonghye Kwon, Kalen Michael, Jiacong Fang, Colin Wong, Zeng Yifu, Diego Montes, et al. ultralytics/yolov5: v6. 2-yolov5 classification models, apple m1, reproducibility, clearml and deci. ai integrations. *Zenodo*, 2022. 1, 2, 6

[13] Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, et al. Yolov6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*, 2022. 1, 2, 6, 7

[14] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *Proceedings of the IEEE/CVF*

Method	O2M-C	O2M-T	MGSA	AP^{val}
RT-DETR	x	x	x	46.5
	✓	x	x	47.4
	x	✓	x	47.5
	x	x	✓	47.5
	✓	✓	✓	**48.1**

表4. 关键组件的消融研究。O2M-C代表基于CNN的一对多辅助分支，O2M-T指基于transformer的一对多密集监督分支，MGSA表示基于transformer的多组自注意力扰动分支。

Number	AP^{val}	AP^{val}_{50}
2	47.9	65.4
3	48.1	65.6
4	48.0	65.3

表5. 自注意力扰动分支数量的消融研究。

自注意力扰动分支数量。为验证自注意力扰动分支数量对RT-DETRv3性能的影响，我们采用RT-DETRv3-R18模型进行消融实验，分别设置分支数为2、3、4，同时保持其他配置不变。如表5所示，当分支数设为3时，模型以48.1的AP值达到最佳性能。减少分支数会降低监督信号的丰富性，导致性能下降；而过度增加分支数则会提升模型学习难度，却未能带来显著的性能提升。

5. 结论

本文提出了一种基于Transformer的实时目标检测算法RT-DETRv3。该算法在RT-DETR基础上，通过引入多重密集正样本辅助监督模块进行改进。这些模块对RT-DETR编码器和解码器的特定特征实施一对多目标监督，从而加速算法收敛并提升性能。需要特别说明的是，这些模块仅用于训练阶段。我们在COCO目标检测基准上验证了算法的有效性，实验表明相比其他实时检测器，本算法取得了更优结果。希望这项工作能为基于Transformer的实时目标检测领域的研究者与开发者带来启发。

参考文献

[1] Alexey Bochkovskiy、Chien-Yao Wang 和 Hong-Yuan Mark Liao。Yolov4：目标检测的最佳速度与精度。*arXiv preprint arXiv:2004.10934*，2020年。1, 2 [2] Nicolas Carion、Francisco Massa、Gabriel Synnaeve、Nicolas Usunier、Alexander Kirillov 和 Sergey Zagoruyko。基于Transformer的端到端目标检测。载于*European confer-ence on computer vision*，第213–229页。Springer，2020年。2 [3] Nicolas Carion、Francisco Massa、Gabriel Synnaeve、Nicolas Usunier、Alexander Kirillov 和 Sergey Zagoruyko。基于Transformer的端到端目标检测。载于*European confer-ence on computer vision*，第213–229页。Springer，2020年。3 [4] 陈强、陈晓康、王健、张山、姚坤、冯浩成、韩俊宇、丁二锐、曾刚和王京东。Group DETR：通过分组一对多分配实现快速DETR训练。载于*Proceedings of the IEEE/CVF International Conference on Computer Vision*，第6633–6642页，2023年。2 [5] 陈玉明、袁新斌、吴瑞琪、王家宝、侯启斌和程明明。YOLO-MS：重新思考实时目标检测中的多尺度表征学习。*arXiv preprint arXiv:2308.05480*，2023年。6, 7 [6] 丁晓晗、张翔宇、马宁宁、韩军功、丁贵广和孙剑。RepVGG：让VGG风格卷积网络重焕光彩。载于*Proceedings of the IEEE/CVF con-ference on computer vision and pattern recognition*，第13733–13742页，2021年。2 [7] 葛政、刘松涛、王峰、李泽明和孙剑。YOLOX：2021年超越YOLO系列。*arXiv preprint arXiv:2107.08430*，2021年。2 [8] 何恺明、张翔宇、任少卿和孙剑。深度残差学习在图像识别中的应用。载于*Proceedings of the IEEE conference on computer vision and pattern recognition*，第770–778页，2016年。3 [9] 胡正东、孙逸凡、王京东和杨毅。DAC-DETR：分治注意力层。*Advances in Neural Information Processing Systems*，36卷，2024年。2 [10] 黄鑫、王欣欣、吕文宇、白雪莹、龙翔、邓凯鹏、党青青、韩树民、刘奇文、胡晓光等。PP-YOLOv2：实用目标检测器。*arXiv preprint arXiv:2104.10419*，2021年。2 [11] Glenn Jocher、Ayush Chaurasia 和 Qiu Jing。Ultralytics YOLO，2023年1月。1, 2, 6, 7 [12] Glenn Jocher、Ayush Chaurasia、Alex Stoken、Jirka Borovec、Yonghye Kwon、Kalen Michael、Jiacong Fang、Colin Wong、Zeng Yifu、Diego Montes等。ultralytics/yolov5：v6.2-YOLOv5分类模型，Apple M1支持，可复现性，ClearML与Deci.ai集成。*Zenodo*，2022年。1, 2, 6 [13] 李楚怡、李露露、姜洪亮、翁凯恒、耿一飞、李亮、柯再丹、李庆元、程梦、聂伟强等。YOLOv6：面向工业应用的单阶段目标检测框架。*arXiv preprintarXiv:2209.02976*，2022年。1, 2, 6, 7 [14] 李峰、张浩、刘世龙、郭健、Ni Lionel M 和张磊。DN-DETR：通过查询去噪加速DETR训练。载于*Proceedings of the IEEE/CVF*

conference on computer vision and pattern recognition, pages 13619–13627, 2022. 2

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 5

[16] 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. *arXiv preprint arXiv:2201.12329*, 2022. 2

[17] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768, 2018. 2

[18] Xiang Long, Kaipeng Deng, Guanzhong Wang, Yang Zhang, Qingqing Dang, Yuan Gao, Hui Shen, Jianguo Ren, Shumin Han, Errui Ding, et al. Pp-yolo: An effective and efficient implementation of object detector. *arXiv preprint arXiv:2007.12099*, 2020. 2

[19] Wenyu Lv, Yian Zhao, Qinyao Chang, Kui Huang, Guanzhong Wang, and Yi Liu. Rt-detrv2: Improved baseline with bag-of-freebies for real-time detection transformer. *arXiv preprint arXiv:2407.17140*, 2024. 2, 5, 6, 7

[20] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 2

[21] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019. 7

[22] Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding. Yolov10: Real-time end-to-end object detection. *arXiv preprint arXiv:2405.14458*, 2024. 1, 6, 7

[23] Chengcheng Wang, Wei He, Ying Nie, Jianyuan Guo, Chuanjian Liu, Yunhe Wang, and Kai Han. Gold-yolo: Efficient object detector via gather-and-distribute mechanism. *Advances in Neural Information Processing Systems*, 36, 2024. 6, 7

[24] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7464–7475, 2023. 1, 2

[25] Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, and I-Hau Yeh. Cspnet: A new backbone that can enhance learning capability of cnn. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 390–391, 2020. 2

[26] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. Yolov9: Learning what you want to learn using programmable gradient information. *arXiv preprint arXiv:2402.13616*, 2024. 1, 2, 6, 7

[27] Chunlong Xia, Xinliang Wang, Feng Lv, Xin Hao, and Yifeng Shi. Vit-comer: Vision transformer with convolutional multi-scale feature interaction for dense predictions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5493–5502, 2024. 2

[28] Shangliang Xu, Xinxin Wang, Wenyu Lv, Qinyao Chang, Cheng Cui, Kaipeng Deng, Guanzhong Wang, Qingqing Dang, Shengyu Wei, Yuning Du, et al. Pp-yoloe: An evolved version of yolo. *arXiv preprint arXiv:2203.16250*, 2022. 3, 4, 5, 6

[29] 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. *arXiv preprint arXiv:2203.03605*, 2022. 2

[30] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9759–9768, 2020. 2

[31] Chuyang Zhao, Yifan Sun, Wenhao Wang, Qiang Chen, Errui Ding, Yi Yang, and Jingdong Wang. Ms-detr: Efficient detr training with mixed supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17027–17036, 2024. 2

[32] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. Detrs beat yolos on real-time object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16965–16974, 2024. 2, 3, 5, 6, 7

[33] Zhuofan Zong, Guanglu Song, and Yu Liu. Detrs with collaborative hybrid assignments training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6748–6758, 2023. 2

conference on computer vision and pattern recognition，第13619–13627页，2022年。2 [15] Tsung-Yi Lin、Michael Maire、Serge Belongie、James Hays、Pietro Perona、Deva Ramanan、Piotr Dollár和C Lawrence Zitnick。Microsoft COCO：上下文中的常见物体。载于*Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*，第740–755页。Springer，2014年。5 [16] Shilong Liu、Feng Li、Hao Zhang、Xiao Yang、Xianbiao Qi、Hang Su、Jun Zhu和Lei Zhang。DAB-DETR：动态锚框是DETR更优的查询。*arXiv preprintarXiv:2201.12329*，2022年。2 [17] Shu Liu、Lu Qi、Haifang Qin、Jianping Shi和Jiaya Jia。用于实例分割的路径聚合网络。载于*Proceedings of the IEEE conference on computer vision and pattern recognition*，第8759–8768页，2018年。2 [18] Xiang Long、Kaipeng Deng、Guanzhong Wang、Yang Zhang、Qingqing Dang、Yuan Gao、Hui Shen、Jianguo Ren、Shumin Han、Errui Ding等。PP-YOLO：一种高效且有效的目标检测器实现。*arXiv preprintarXiv:2007.12099*，2020年。2 [19] Wenyu Lv、Yian Zhao、Qinyao Chang、Kui Huang、Guanzhong Wang和Yi Liu。RT-DETRv2：通过免费技巧包改进的实时检测Transformer基线。*arXiv preprint arXiv:2407.17140*，2024年。2, 5, 6, 7 [20] Shaoqing Ren、Kaiming He、Ross Girshick和Jian Sun。Faster R-CNN：利用区域提议网络实现实时目标检测。*Advances in neural information process-ing systems*，28，2015年。2 [21] Shuai Shao、Zeming Li、Tianyuan Zhang、Chao Peng、Gang Yu、Xiangyu Zhang、Jing Li和Jian Sun。Objects365：用于目标检测的大规模高质量数据集。载于*Proceedings of the IEEE/CVF international conference on computer vision*，第8430–8439页，2019年。7 [22] Ao Wang、Hui Chen、Lihao Liu、Kai Chen、Zijia Lin、Jungong Han和Guiguang Ding。YOLOv10：实时端到端目标检测。*arXiv preprint arXiv:2405.14458*，2024年。1, 6, 7 [23] Chengcheng Wang、Wei He、Ying Nie、Jianyuan Guo、Chuanjian Liu、Yunhe Wang和Kai Han。Gold-YOLO：通过收集-分发机制实现的高效目标检测器。*Advances in Neural Information Processing Systems*，36，2024年。6, 7 [24] Chien-Yao Wang、Alexey Bochkovskiy和Hong-Yuan Mark Liao。YOLOv7：可训练免费技巧包为实时目标检测器设定新标杆。载于*Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*，第7464–7475页，2023年。1, 2 [25] Chien-Yao Wang、Hong-Yuan Mark Liao、Yueh-Hua Wu、Ping-Yang Chen、Jun-Wei Hsieh和I-Hau Yeh。CSPNet：一种能增强CNN学习能力的新骨干网络。载于*Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*，第390–391页，2020年。2 [26] Chien-Yao Wang、I-Hau Yeh和Hong-Yuan Mark Liao。YOLOv9：利用可编程梯度信息学习你想学的内容。*arXiv preprintarXiv:2402.13616*，2024年。1, 2, 6, 7 [27] 夏春龙，王新亮，吕峰，郝鑫，石一峰。Vit-comer：用于密集预测的卷积多尺度特征交互视觉Transformer。载于*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*，第5493–5502页，2024年。2 [28] 徐尚亮，王欣欣，吕文宇，常钦尧，崔程，邓凯鹏，王冠中，党青青，魏胜宇，杜宇宁等。PP-YOLOE：YOLO的进化版本。*arXiv preprint arXiv:2203.16250*，2022年。3, 4, 5, 6 [29] 张浩，李峰，刘世龙，张磊，苏航，朱军，倪立昂，沈向洋。DINO：基于改进去噪锚框的端到端目标检测DETR模型。*arXiv preprint arXiv:2203.03605*，2022年。2 [30] 张世峰，迟程，姚永强，雷震，李世章。通过自适应训练样本选择弥合基于锚与无锚检测的差距。载于*Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*，第9759–9768页，2020年。2 [31] 赵楚阳，孙一凡，王文浩，陈强，丁二瑞，杨毅，王京东。MS-DETR：混合监督下的高效DETR训练。载于*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*，第17027–17036页，2024年。2 [32] 赵一安，吕文宇，徐尚亮，魏金满，王冠中，党青青，刘毅，陈杰。实时目标检测中DETR超越YOLO。载于*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*，第16965–16974页，2024年。2, 3, 5, 6, 7 [33] 宗卓凡，宋广录，刘宇。协作混合分配训练的DETR模型。载于*Proceedings of the IEEE/CVF international conference on computer vision*，第6748–6758页，2023年。2