DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection

Hao Zhang[1,3*†], Feng Li[1,3*†], Shilong Liu[2,3*†], Lei Zhang[3‡],
Hang Su[2], Jun Zhu[2], Lionel M. Ni[1,4], Heung-Yeung Shum[1,3]

[1]The Hong Kong University of Science and Technology.
[2]Dept. of CST., BNRist Center, Institute for AI, Tsinghua University.
[3]International Digital Economy Academy (IDEA).
[4]The Hong Kong University of Science and Technology (Guangzhou).
{hzhangcx,fliay}@connect.ust.hk
{liusl20}@mails.tsinghua.edu.cn
{suhangss,dcszj}@mail.tsinghua.edu.cn
{ni,hshum}@ust.hk
{leizhang}@idea.edu.cn

Abstract. We present DINO (**D**ETR with **I**mproved de**N**oising anch**O**r boxes), a state-of-the-art end-to-end object detector. DINO improves over previous DETR-like models in performance and efficiency by using a contrastive way for denoising training, a mixed query selection method for anchor initialization, and a look forward twice scheme for box prediction. DINO achieves 49.4AP in 12 epochs and 51.3AP in 24 epochs on COCO with a ResNet-50 backbone and multi-scale features, yielding a significant improvement of **+6.0AP** and **+2.7AP**, respectively, compared to DN-DETR, the previous best DETR-like model. DINO scales well in both model size and data size. Without bells and whistles, after pre-training on the Objects365 dataset with a SwinL backbone, DINO obtains the best results on both COCO `val2017` (**63.2AP**) and `test-dev` (**63.3AP**). Compared to other models on the leaderboard, DINO significantly reduces its model size and pre-training data size while achieving better results. Our code will be available at https://github.com/IDEACVR/DINO.

Keywords: Object Detection; Detection Transformer; End-to-End Detector

1 Introduction

Object detection is a fundamental task in computer vision. Remarkable progress has been accomplished by classical convolution-based object detection algo-

* Equal contribution. Listing order is random.
† This work was done when Hao Zhang, Feng Li, and Shilong Liu were interns at IDEA.
‡ Corresponding author.

DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection

张浩[1,3*†]，李锋[1,3*†]，刘世龙[2,3*†]，张磊[3‡]，苏航[2]，朱军[2]，倪明选[1,4]，沈向洋[1,3]

[1]香港科技大学。[2]清华大学计算机科学与技术系，北京国家信息科学技术研究中心，人工智能研究院。[3]国际数字经济学院（IDEA）。[4]香港科技大学（广州）。{hzhangcx,fliay}@connect.ust.hk {liusl20}@mails.tsinghua.edu.cn {suhangss,dcszj}@mail.tsinghua.edu.cn {ni,hshum}@ust.hk {leizhang}@idea.edu.cn

Abstract. 我们提出了DINO（**D**ETR结合**I**改进的去**N**噪锚**O**框方案），一种最先进的端到端目标检测器。DINO通过采用对比式去噪训练方法、混合查询选择的锚框初始化策略以及前瞻两次的边界框预测机制，在性能与效率上超越了以往的类DETR模型。在ResNet-50骨干网络与多尺度特征加持下，DINO仅用12个周期便达到49.4AP，24个周期提升至51.3AP（COCO数据集），相较此前最优的类DETR模型DN-DETR分别实现了**+6.0AP**和**+2.7AP**的显著提升。DINO在模型规模与数据规模上均展现出优异扩展性：基于SwinL骨干网络在Objects365数据集预训练后，无需复杂技巧即可在COCO val2017（**63.2AP**）和test-dev（**63.3AP**）上取得最佳成绩。相比榜单其他模型，DINO在显著减小模型规模与预训练数据量的同时，仍能获得更优性能。代码将发布于https://github.com/IDEACVR/DINO。

Keywords: 目标检测; 检测变换器; 端到端检测器

1 Introduction

物体检测是计算机视觉中的一项基础任务。基于经典卷积的物体检测算法已取得了显著进展。

* Equal contribution. Listing order is random.

† This work was done when Hao Zhang, Feng Li, and Shilong Liu were interns at IDEA.

‡ Corresponding author.

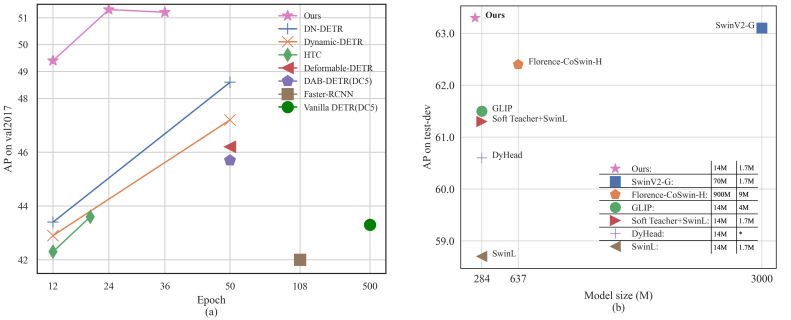

Fig. 1. AP on COCO compared with other detection models. (a) Comparison to models with a ResNet-50 backbone w.r.t. training epochs. Models marked with DC5 use a dilated larger resolution feature map. Other models use multi-scale features. (b) Comparison to SOTA models w.r.t. pre-training data size and model size. SOTA models are from the COCO `test-dev` leaderboard. In the legend we list the backbone pre-training data size (first number) and detection pre-training data size (second number). $*$ means the data size is not disclosed.

rithms [31,35,19,2,12]. Despite that such algorithms normally include hand-designed components like anchor generation and non-maximum suppression (NMS), they yield the best detection models such as DyHead [7], Swin [23] and SwinV2 [22] with HTC++ [4], as evidenced on the COCO test-dev leaderboard [1].

In contrast to classical detection algorithms, DETR [3] is a novel Transformer-based detection algorithm. It eliminates the need of hand-designed components and achieves comparable performance with optimized classical detectors like Faster RCNN [31]. Different from previous detectors, DETR models object detection as a set prediction task and assigns labels by bipartite graph matching. It leverages learnable queries to probe the existence of objects and combine features from an image feature map, which behaves like soft ROI pooling [21].

Despite its promising performance, the training convergence of DETR is slow and the meaning of queries is unclear. To address such problems, many methods have been proposed, such as introducing deformable attention [41], decoupling positional and content information [25], providing spatial priors [11,39,37], etc. Recently, DAB-DETR [21] proposes to formulate DETR queries as dynamic anchor boxes (DAB), which bridges the gap between classical anchor-based detectors and DETR-like ones. DN-DETR [17] further solves the instability of bipartite matching by introducing a denoising (DN) technique. The combination of DAB and DN makes DETR-like models competitive with classical detectors on both training efficiency and inference performance.

The best detection models nowadays are based on improved classical detectors like DyHead [8] and HTC [4]. For example, the best result presented in SwinV2 [22] was trained with the HTC++ [4,23] framework. Two main reasons contribute to the phenomenon: 1) *Previous DETR-like models are inferior* to the improved classical detectors. Most classical detectors have been well studied and

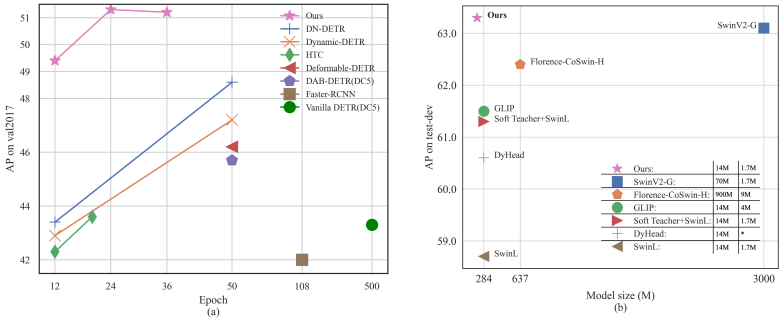

Fig. 1. 在COCO数据集上的AP与其他检测模型的比较。(a) 与采用ResNet-50骨干网络的模型在不同训练周期下的对比。标记为DC5的模型使用了扩张的更高分辨率特征图，其他模型则采用多尺度特征。(b) 与SOTA模型在预训练数据规模和模型大小方面的对比。SOTA模型来自COCO test-dev排行榜。图例中我们列出了骨干网络预训练数据规模（第一个数字）和检测任务预训练数据规模（第二个数字）。∗表示数据规模未公开。

算法[31,35,19,2,12]。尽管这类算法通常包含手工设计的组件，如锚点生成和非极大值抑制（NMS），但它们产出了最佳的检测模型，例如DyHead[7]、Swin[23]和SwinV2[22]结合HTC++[4]，这一点在COCO test-dev排行榜[1]上得到了验证。

与经典检测算法不同，DETR[3]是一种基于Transformer的新型检测算法。它摒弃了手工设计的组件，性能与经过优化的经典检测器（如Faster RCNN[31]）相当。不同于以往的检测器，DETR将目标检测建模为集合预测任务，并通过二分图匹配分配标签。该算法利用可学习的查询向量来探测目标存在性，并融合图像特征图中的特征，其作用类似于软性ROI池化[21]。

尽管DETR表现前景广阔，但其训练收敛速度较慢且查询(query)的含义不明确。为解决这些问题，研究者提出了多种改进方法，如引入可变形注意力机制[41]、解耦位置与内容信息[25]、提供空间先验[11,39,37]等。近期，DAB-DETR[21]创新性地将DETR查询表述为动态锚框(DAB)，弥合了传统基于锚点的检测器与类DETR检测器之间的鸿沟。DN-DETR[17]则通过引入去噪(DN)技术，进一步解决了二分匹配不稳定的问题。DAB与DN的结合使类DETR模型在训练效率和推理性能上均能与传统检测器媲美。

现今最佳的检测模型基于改进的经典检测器，如DyHead [8]和HTC [4]。例如，SwinV2 [22]中展示的最佳结果便是采用HTC++ [4,23]框架训练所得。这一现象主要由两个原因促成：1) *Previous DETR-like models are inferior* 对改进经典检测器的贡献。大多数经典检测器已得到深入研究，且

highly optimized, leading to a better performance compared with the newly developed DETR-like models. For instance, the best performing DETR-like models nowadays are still under 50 AP on COCO. 2) The *scalability* of DETR-like models has not been well studied. There is no reported result about how DETR-like models perform when scaling to a large backbone and a large-scale data set. We aim to address both concerns in this paper.

Specifically, by improving the denoising training, query initialization, and box prediction, we design a new DETR-like model based on DN-DETR [17], DAB-DETR [21], and Deformable DETR [41]. We name our model as **DINO** (**D**ETR with **I**mproved de**N**oising anch**O**r box). As shown in Fig. 1, the comparison on COCO shows the superior performance of DINO. In particular, DINO demonstrates a great scalability, setting a new record of 63.3 AP on the COCO test-dev leaderboard [1]

As a DETR-like model, DINO contains a backbone, a multi-layer Transformer encoder, a multi-layer Transformer decoder, and multiple prediction heads. Following DAB-DETR [21], we formulate queries in decoder as dynamic anchor boxes and refine them step-by-step across decoder layers. Following DN-DETR [17], we add ground truth labels and boxes with noises into the Transformer decoder layers to help stabilize bipartite matching during training. We also adopt deformable attention [41] for its computational efficiency. Moreover, we propose three new methods as follows. First, to improve the one-to-one matching, we propose a *contrastive denoising training* by adding both positive and negative samples of the same ground truth at the same time. After adding two different noises to the same ground truth box, we mark the box with a smaller noise as positive and the other as negative. The contrastive denoising training helps the model to avoid duplicate outputs of the same target. Second, the dynamic anchor box formulation of queries links DETR-like models with classical two-stage models. Hence we propose a *mixed query selection* method, which helps better initialize the queries. We select initial anchor boxes as positional queries from the output of the encoder, similar to [41,39]. However, we leave the content queries learnable as before, encouraging the first decoder layer to focus on the spatial prior. Third, to leverage the refined box information from later layers to help optimize the parameters of their adjacent early layers, we propose a new *look forward twice* scheme to correct the updated parameters with gradients from later layers.

We validate the effectiveness of DINO with extensive experiments on the COCO [20] detection benchmarks. As shown in Fig. 1, DINO achieves 49.4AP in 12 epochs and 51.3AP in 24 epochs with ResNet-50 and multi-scale features, yielding a significant improvement of **+6.0**AP and **+2.7**AP, respectively, compared to the previous best DETR-like model. In addition, DINO scales well in both model size and data size. After pre-training on the Objects365 [33] data set with a SwinL [23] backbone, DINO achieves the best results on both COCO `val2017` (**63.2**AP) and `test-dev` (**63.3**AP) benchmarks, as shown in Table 3. Compared to other models on the leaderboard [1], we reduce the model size to **1/15** compared to SwinV2-G [22]. Compared to Florence [40], we reduce the

经过高度优化，相比新开发的类DETR模型具有更优性能。例如，当前性能最佳的类DETR模型在COCO数据集上的AP仍低于50。2）类DETR模型的*scalability*尚未得到充分研究。目前未见关于类DETR模型在大型骨干网络和大规模数据集上扩展性能的公开报道。本文旨在同时解决这两个问题。

具体而言，通过改进去噪训练、查询初始化和框预测，我们在DN-DETR[17]、DAB-DETR[21]和Deformable DETR[41]的基础上设计了一个新的类DETR模型。我们将该模型命名为**DINO**(**D**ETR，其具备**I**改进的去**N**噪锚**O**框)特性。如图1所示，在COCO数据集上的对比实验表明DINO具有卓越性能。尤为突出的是，DINO展现出极强的可扩展性，在COCO test-dev排行榜[1]上以63.3 AP的成绩刷新了记录。

作为一种类似DETR的模型，DINO包含一个主干网络、一个多层Transformer编码器、一个多层Transformer解码器以及多个预测头。遵循DAB-DETR[21]的做法，我们将解码器中的查询表述为动态锚框，并逐步在解码器层间进行细化。参照DN-DETR[17]的方法
，
我们在Transformer解码器层中加入带有噪声的真实标签和边界框，以帮助稳定训练期间的二分匹配。同时，为了计算效率，我们采用了可变形注意力机制[41]。此外，我们还提出了以下三种新方法。首先，为了改善一对一匹配，我们提出了一种*contrastive denoising training*方法，即同时加入同一真实样本的正负样本。对同一真实边界框添加两种不同噪声后，我们将噪声较小的标记为正样本，另一为负样本。这种对比去噪训练有助于模型避免对同一目标产生重复输出。其次，查询的动态锚框构建将类DETR模型与经典两阶段模型联系起来。因此我们提出*mixed query selection*方法，以更好地初始化查询。与[41,39]类似，我们从编码器输出中选择初始锚框作为位置查询，但保持内容查询可学习不变，促使第一解码器层聚焦于空间先验。第三，为了利用深层提供的精修框信息来优化相邻浅层参数，我们提出全新的*lookforward twice*方案，通过来自深层的梯度修正更新参数。

我们在COCO[20]检测基准上通过大量实验验证了DINO的有效性。如图1所示，采用ResNet-50骨干网络和多尺度特征时，DINO在12个训练周期内达到49.4 AP，24个周期提升至51.3AP，相较之前最优的类DETR模型分别实现了$+\mathbf{6.0}$AP和$+\mathbf{2.7}$AP的显著提升。此外，DINO在模型规模与数据规模上均展现出良好的扩展性。使用SwinL[23]骨干网络在Objects365[33]数据集上进行预训练后，DINO在COCO val2017（**63.2**AP）和test-dev（**63.3**AP）基准上均取得最佳成绩（见表3）。相较于排行榜[1]上其他模型，我们将模型体积缩减至SwinV2-G[22]的$\mathbf{1/15}$；相比Florence[40]，我们进一步降低了

pre-training detection dataset to **1/5** and backbone pre-training dataset to **1/60** while achieving better results.

We summarize our contributions as follows.

1. We design a new end-to-end DETR-like object detector with several novel techniques, including contrastive DN training, mixed query selection, and look forward twice for different parts of the DINO model.
2. We conduct intensive ablation studies to validate the effectiveness of different design choices in DINO. As a result, DINO achieves 49.4AP in 12 epochs and 51.3AP in 24 epochs with ResNet-50 and multi-scale features, significantly outperforming the previous best DETR-like models. In particular, DINO trained in 12 epochs shows a more significant improvement on small objects, yielding an improvement of **+7.5**AP.
3. We show that, without bells and whistles, DINO can achieve the best performance on public benchmarks. After pre-training on the Objects365 [33] dataset with a SwinL [23] backbone, DINO achieves the best results on both COCO `val2017` (**63.2**AP) and `test-dev` (**63.3**AP) benchmarks. To the best of our knowledge, this is the first time that an end-to-end Transformer detector outperforms state-of-the-art (SOTA) models on the COCO leaderboard [1].

2 Related Work

2.1 Classical Object Detectors

Early convolution-based object detectors are either two-stage or one-stage models, based on hand-crafted anchors or reference points. Two-stage models [30,13] usually use an region proposal network (RPN) [30] to propose potential boxes, which are then refined in the second stage. One-stage models such as YOLO v2 [28] and YOLO v3 [29] directly output offsets relative to predefined anchors. Recently, some convolution-based models such as HTC++ [4] and Dyhead [7] have achieved top performance on the COCO 2017 dataset [20]. The performance of convolution-based models, however, relies on the way they generate anchors. Moreover, they need hand-designed components like NMS to remove duplicate boxes, and hence cannot perform end-to-end optimization.

2.2 DETR and Its Variants

Carion *et al.* [3] proposed a Transformer-based end-to-end object detector named DETR (DEtection TRansformer) without using hand-designed components like anchor design and NMS. Many follow-up papers have attempted to address the slow training convergence issue of DETR introduced by decoder cross-attention. For instance, Sun *et al.* [34] designed an encoder-only DETR without using a decoder. Dai *et al.* [7] proposed a dynamic decoder to focus on important regions from multiple feature levels.

预训练检测数据集设为**1/5**，骨干网络预训练数据集设为**1/60**，同时取得更好的结果。

我们将贡献总结如下。

1. 我们设计了一种新型端到端类DETR目标检测器，融入了多项创新技术，包括对比式DN训练、混合查询选择以及针对DINO模型不同部分的二次前瞻机制。
2. 我们进行了深入的消融研究，以验证DINO中不同设计选择的有效性。结果表明，采用ResNet-50和多尺度特征的DINO在12个训练周期内达到49.4AP，24个周期内提升至51.3AP，显著超越了之前最佳的类DETR模型。特别值得注意的是，经过12周期训练的DINO在小物体检测上展现出更为显著的提升，实现了$+$**7.5**AP的性能改进。
3. 我们证明，无需任何花哨的技巧，DINO就能在公开基准测试中取得最佳性能。采用SwinL[23]骨干网络在Objects365[33]数据集上进行预训练后，DINO在COCO val2017（**63.2**AP）和test-dev（**63.3**AP）基准测试中均获得了最优结果。据我们所知，这是首个端到端Transformer检测器在COCO排行榜[1]上超越最先进（SOTA）模型的案例。

2 Related Work

2.1 Classical Object Detectors

早期的基于卷积的目标检测器主要分为两阶段或单阶段模型，它们依赖于手工设计的锚框或参考点。两阶段模型[30,13]通常利用区域提议网络(RPN)[30]生成潜在候选框，随后在第二阶段进行精细化调整。而单阶段模型如YOLO v2[28]和YOLO v3[29]则直接输出相对于预定义锚框的偏移量。近期，诸如HTC++[4]和Dyhead[7]等基于卷积的模型在COCO 2017数据集[20]上取得了顶尖性能。然而，这类模型的性能高度依赖于锚框生成方式，且需依赖非极大值抑制(NMS)等人工设计组件来消除重复检测框，因而无法实现端到端优化。

2.2 DETR and Its Variants

Carion *et al.* [3] 提出了一种基于Transformer的端到端目标检测器DETR（DEtection TRansformer），无需使用人工设计的组件如锚框和非极大值抑制（NMS）。许多后续研究尝试解决由解码器交叉注意力引起的DETR训练收敛慢的问题。例如，Sun *et al.* [34] 设计了一种仅含编码器的DETR，省去了解码器部分。Dai *et al.* [7] 则提出了一种动态解码器，能够聚焦于多层级特征中的重要区域。

Another line of works is towards a deeper understanding of decoder queries in DETR. Many papers associate queries with spatial position from different perspectives. Deformable DETR [41] predicts 2D anchor points and designs a deformable attention module that only attends to certain sampling points around a reference point. Efficient DETR [39] selects top K positions from encoder's dense prediction to enhance decoder queries. DAB-DETR [21] further extends 2D anchor points to 4D anchor box coordinates to represent queries and dynamically update boxes in each decoder layer. Recently, DN-DETR [17] introduces a denoising training method to speed up DETR training. It feeds noise-added ground-truth labels and boxes into the decoder and trains the model to reconstruct the original ones. Our work of DINO in this paper is based on DAB-DETR and DN-DETR, and also adopts deformable attention for its computational efficiency.

2.3 Large-scale Pre-training for Object Detection

Large-scale pre-training have had a big impact on both natural language processing [10] and computer vision [27]. The best performance detectors nowadays are mostly achieved with large backbones pre-trained on large-scale data. For example, Swin V2 [22] extends its backbone size to 3.0 Billion parameters and pre-trains its models with 70M privately collected images. Florence [40] first pre-trains its backbone with 900M privately curated image-text pairs and then pre-trains its detector with 9M images with annotated or pseudo boxes. In contrast, DINO achieves the SOTA result with a publicly available SwinL [23] backbone and a public dataset Objects365 [33] (1.7M annotated images) only.

3 DINO: DETR with Improved DeNoising Anchor Boxes

3.1 Preliminaries

As studied in Conditional DETR [25] and DAB-DETR [21], it becomes clear that queries in DETR [3] are formed by two parts: a positional part and a content part, which are referred to as positional queries and content queries in this paper. DAB-DETR [21] explicitly formulates each positional query in DETR as a 4D anchor box (x, y, w, h), where x and y are the center coordinates of the box and w and h correspond to its width and height. Such an explicit anchor box formulation makes it easy to dynamically refine anchor boxes layer by layer in the decoder.

DN-DETR [17] introduces a denoising (DN) training method to accelerate the training convergence of DETR-like models. It shows that the slow convergence problem in DETR is caused by the instability of bipartite matching. To mitigate this problem, DN-DETR proposes to additionally feed noised ground-truth (GT) labels and boxes into the Transformer decoder and train the model to reconstruct the ground-truth ones. The added noise $(\Delta x, \Delta y, \Delta w, \Delta h)$ is constrained by $|\Delta x| < \frac{\lambda w}{2}$, $|\Delta y| < \frac{\lambda h}{2}$, $|\Delta w| < \lambda w$, and $|\Delta y| < \lambda h$, where (x, y, w, h) denotes

另一研究方向致力于更深入理解DETR中的解码器查询机制。多篇论文从不同视角将查询与空间位置关联：可变形DETR[41]预测二维锚点并设计可变形注意力模块，使其仅关注参考点周围的特定采样位置；高效DETR[39]从编码器的密集预测中筛选Top K位置来增强解码器查询；DAB-DETR[21]进一步将二维锚点扩展为四维锚框坐标来表示查询，并在每个解码层动态更新检测框。最近，DN-DETR[17]提出去噪训练方法以加速DETR训练，其向解码器输入添加噪声的真实标签与检测框，并训练模型重建原始目标。本文提出的DINO方法基于DAB-DETR和DN-DETR框架，同时采用计算高效的可变形注意力机制。

2.3 Large-scale Pre-training for Object Detection

大规模预训练对自然语言处理[10]和计算机视觉[27]领域都产生了深远影响。当前性能最优的检测器大多采用基于海量数据预训练的大型骨干网络。例如，Swin V2[22]将其骨干网络参数量扩展至3.0亿，并利用7000万张私有收集图像进行模型预训练。Florence[40]则首先使用9亿组私有整理的图文对预训练骨干网络，再基于900万张带标注或伪标注框的图像预训练检测器。相比之下，DINO仅采用公开可用的SwinL[23]骨干网络和Objects365[33]公开数据集（170万张标注图像），就达到了当前最先进的性能水平。

3 DINO: DETR with Improved DeNoising Anchor Boxes

3.1 Preliminaries

如Conditional DETR[25]和DAB-DETR[21]中所研究的那样，可以清楚地看到DETR[3]中的查询由两部分组成：位置部分和内容部分，在本文中分别称为位置查询和内容查询。DAB-DETR[21]明确将DETR中的每个位置查询表述为一个4D锚框(x, y, w, h)，其中x和y表示框的中心坐标，w和h对应其宽度和高度。这种显式的锚框表述使得在解码器中逐层动态优化锚框变得容易。

DN-DETR [17] 提出了一种去噪（DN）训练方法，以加速类DETR模型的训练收敛。研究表明，DETR中收敛缓慢的问题源于二分图匹配的不稳定性。为解决这一问题，DN-DETR建议额外向Transformer解码器输入带有噪声的真实（GT）标签和边界框，并训练模型重建原始真实值。所添加的噪声$\Delta x, \Delta y, \Delta w, \Delta h$受到$|\Delta x| < \frac{\lambda w}{2}$、$|\Delta y| < \frac{\lambda h}{2}$、$|\Delta w| < \lambda w$和$|\Delta y| < \lambda h$的约束，其中$(x, y, w, h)$表示

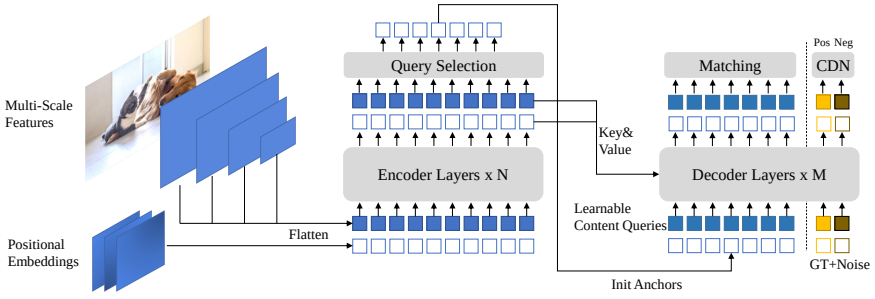

Fig. 2. The framework of our proposed DINO model. Our improvements are mainly in the Transformer encoder and decoder. The top-K encoder features in the last layer are selected to initialize the positional queries for the Transformer decoder, whereas the content queries are kept as learnable parameters. Our decoder also contains a Contrastive DeNoising (CDN) part with both positive and negative samples.

a GT box and λ[1] is a hyper-parameter to control the scale of noise. Since DN-DETR follows DAB-DETR to view decoder queries as anchors, a noised GT box can be viewed as a special anchor with a GT box nearby as λ is usually small. In addition to the orginal DETR queries, DN-DETR adds a DN part which feeds noised GT labels and boxes into the decoder to provide an auxiliary DN loss. The DN loss effectively stabilizes and speeds up the DETR training and can be plugged into any DETR-like models.

Deformable DETR [41] is another early work to speed up the convergence of DETR. To compute deformable attention, it introduces the concept of reference point so that deformable attention can attend to a small set of key sampling points around a reference. The reference point concept makes it possible to develop several techniques to further improve the DETR performance. The first technique is query selection[2], which selects features and reference boxes from the encoder as inputs to the decoder directly. The second technique is iterative bounding box refinement with a careful gradient detachment design between two decoder layers. We call this gradient detachment technique "look forward once" in our paper.

Following DAB-DETR and DN-DETR, DINO formulates the positional queries as dynamic anchor boxes and is trained with an extra DN loss. Note that DN-DETR also adopts several techniques from Deformable DETR to achieve a better performance, including its deformable attention mechanism and "look forward

[1] The DN-DETR paper [17] uses λ_1 and λ_2 to denote noise scales of center shifting and box scaling, but sets $\lambda_1 = \lambda_2$. In this paper, we use λ in place of $\lambda_1$andλ_2 for simplicity.

[2] Also named as "two-stage" in the Deformable DETR paper. As the "two-stage" name might confuse readers with classical detectors, we use the term "query selection" instead in our paper.

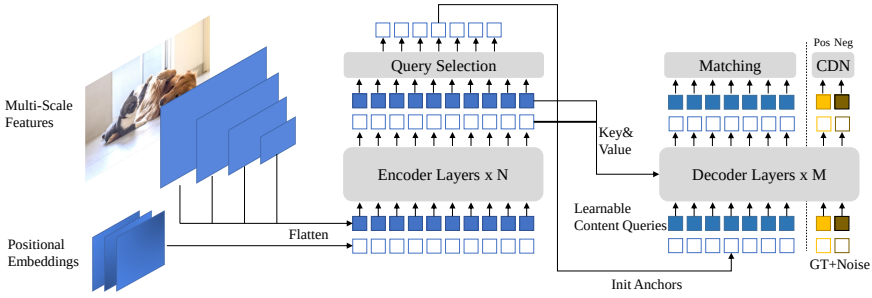

Fig. 2. 我们提出的DINO模型框架。我们的改进主要集中在Transformer编码器与解码器部分。通过选取最后一层中的top-K编码器特征来初始化Transformer解码器的位置查询，而内容查询则保持为可学习参数。解码器还包含一个对比去噪（CDN）模块，该模块同时处理正负样本。

一个GT框，λ[1]是控制噪声尺度的超参数。由于DN-DETR遵循DAB-DETR将解码器查询视为锚点，带有噪声的GT框可视为一种特殊锚点，其附近存在一个GT框，因为λ通常较小。除了原始的DETR查询外，DN-DETR还增加了一个DN部分，将带有噪声的GT标签和框输入解码器，以提供辅助的DN损失。该DN损失有效稳定并加速了DETR训练，且可嵌入任何类DETR模型中。

可变形DETR[41]是另一项加速DETR收敛的早期工作。为计算可变形注意力，它引入了参考点的概念，使得可变形注意力能够聚焦于参考点周围的一小组关键采样点。参考点概念的提出为开发多项技术以进一步提升DETR性能奠定了基础。第一项技术是查询选择[2]，该技术直接从编码器中选取特征和参考框作为解码器的输入。第二项技术是通过在两层解码器间精心设计的梯度分离机制实现迭代边界框优化。我们在论文中将这种梯度分离技术称为"前瞻一次"。

继DAB-DETR和DN-DETR之后，DINO将位置查询构建为动态锚框，并通过额外的DN损失进行训练。值得注意的是，DN-DETR还采用了Deformable DETR的多项技术以提升性能，包括其可变形注意力机制及“前瞻”策略。

[1] The DN-DETR paper [17] uses λ_1 and λ_2 to denote noise scales of center shifting and box scaling, but sets $\lambda_1 = \lambda_2$. In this paper, we use λ in place of $\lambda_1$andλ_2 for simplicity.

[2] Also named as “two-stage” in the Deformable DETR paper. As the “two-stage” name might confuse readers with classical detectors, we use the term “query selection” instead in our paper.

once" implementation in layer parameter update. DINO further adopts the query selection idea from Deformable DETR to better initialize the positional queries. Built upon this strong baseline, DINO introduces three novel methods to further improve the detection performance, which will be described in Sec. 3.3, Sec. 3.4, and Sec. 3.5, respectively.

3.2 Model Overview

As a DETR-like model, DINO is an end-to-end architecture which contains a backbone, a multi-layer Transformer [36] encoder, a multi-layer Transformer decoder, and multiple prediction heads. The overall pipeline is shown in Fig. 2. Given an image, we extract multi-scale features with backbones like ResNet [14] or Swin Transformer [23], and then feed them into the Transformer encoder with corresponding positional embeddings. After feature enhancement with the encoder layers, we propose a new mixed query selection strategy to initialize anchors as positional queries for the decoder. Note that this strategy does not initialize content queries but leaves them learnable. More details of mixed query selection are available in Sec. 3.4. With the initialized anchors and the learnable content queries, we use the deformable attention [41] to combine the features of the encoder outputs and update the queries layer-by-layer. The final outputs are formed with refined anchor boxes and classification results predicted by refined content features. As in DN-DETR [17], we have an extra DN branch to perform denoising training. Beyond the standard DN method, we propose a new contrastive denoising training approach by taking into account hard negative samples, which will be presented in Sec. 3.3. To fully leverage the refined box information from later layers to help optimize the parameters of their adjacent early layer, a novel look forward twice method is proposed to pass gradients between adjacent layers, which will be described in Sec. 3.5.

3.3 Contrastive DeNoising Training

DN-DETR is very effective in stabilizing training and accelerating convergence. With the help of DN queries, it learns to make predictions based on anchors which have GT boxes nearby. However, it lacks a capability of predicting "no object" for anchors with no object nearby. To address this issue, we propose a Contrastive DeNoising (CDN) approach to *rejecting* useless anchors.

Implementation: DN-DETR has a hyper-parameter λ to control the noise scale. The generated noises are no larger than λ as DN-DETR wants the model to reconstruct the ground truth (GT) from moderately noised queries. In our method, we have two hyper-parameters λ_1 and λ_2, where $\lambda_1 < \lambda_2$. As shown in the concentric squares in Fig. 3, we generate two types of CDN queries: positive queries and negative queries. Positive queries within the inner square have a noise scale smaller than λ_1 and are expected to reconstruct their corresponding ground truth boxes. Negative queries between the inner and outer squares have a noise scale larger than λ_1 and smaller than λ_2. They are are expected to predict "no object". We usually adopt small λ_2 because hard negative samples closer to

DINO在层参数更新中采用了“once”实现方式，并进一步借鉴了Deformable DETR的查询选择思想，以更好地初始化位置查询。基于这一强大基线，DINO引入了三种新颖方法来进一步提升检测性能，这些方法将分别在3.3节、3.4节和3.5节中详细阐述。

3.2 Model Overview

作为一款类DETR模型，DINO采用端到端架构，包含主干网络、多层Transformer[36]编码器、多层Transformer解码器及多个预测头。整体流程如图2所示。给定输入图像，我们通过ResNet[14]或Swin Transformer[23]等主干网络提取多尺度特征，并配合位置嵌入输入Transformer编码器。经编码层特征增强后，我们提出混合查询选择新策略，将初始化锚点作为解码器的位置查询。需注意该策略不初始化内容查询，而保持其可学习性。混合查询选择细节详见3.4节。基于初始化锚点与可学习内容查询，我们采用可变形注意力[41]融合编码器输出特征，逐层更新查询。最终输出由精炼锚框与基于优化内容特征预测的分类结果构成。如DN-DETR[17]所示，我们增设去噪分支进行降噪训练。除标准DN方法外，我们还提出考虑困难负样本的对比式降噪训练新方法，详见3.3节。为充分利用深层精炼框信息优化相邻浅层参数，创新性提出跨层梯度传递的双重前瞻方法，具体实现见3.5节。

3.3 Contrastive DeNoising Training

DN-DETR在稳定训练和加速收敛方面非常有效。借助DN查询，它学会了基于附近有GT框的锚点进行预测。然而，它缺乏对附近没有物体的锚点预测“无物体”的能力。为了解决这个问题，我们提出了一种对比去噪（CDN）方法来*rejecting*无用的锚点。

Implementation: DN-DETR有一个超参数λ用于控制噪声尺度。生成的噪声不超过λ，因为DN-DETR希望模型能从适度噪声化的查询中重建真实标注(GT)。在我们的方法中，我们有两个超参数λ_1和λ_2，其中$\lambda_1 < \lambda_2$。如图3中的同心方块所示，我们生成两种类型的CDN查询：正查询和负查询。内方块内的正查询噪声尺度小于λ_1，预期能重建其对应的真实标注框。内外方块之间的负查询噪声尺度大于λ_1但小于λ_2，它们预期预测为“无物体”。我们通常采用较小的λ_2，因为更接近的困难负样本

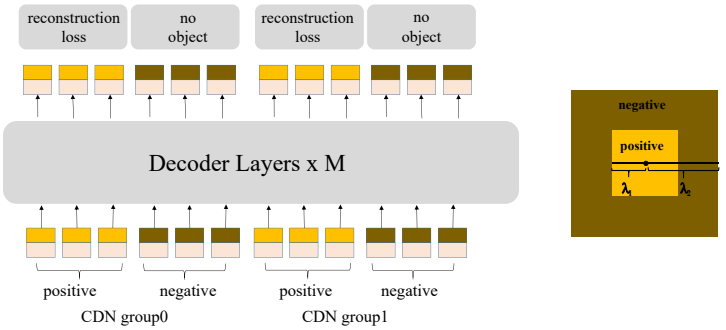

Fig. 3. The structure of CDN group and a demonstration of positive and negative examples. Although both positive and negative examples are 4D anchors that can be represented as points in 4D space, we illustrate them as points in 2D space on concentric squares for simplicity. Assuming the square center is a GT box, points inside the inner square are regarded as a positive example and points between the inner square and the outer square are viewed as negative examples.

GT boxes are more helpful to improve the performance. As shown in Fig. 3, each CDN group has a set of positive queries and negative queries. If an image has n GT boxes, a CDN group will have $2 \times n$ queries with each GT box generating a positive and a negative queries. Similar to DN-DETR, we also use multiple CDN groups to improve the effectiveness of our method. The reconstruction losses are l_1 and GIOU losses for box regression and focal loss [19] for classification. The loss to classify negative samples as background is also focal loss.

Analysis: The reason why our method works is that it can inhibit confusion and select high-quality anchors (queries) for predicting bounding boxes. The confusion happens when multiple anchors are close to one object. In this case, it is hard for the model to decide which anchor to choose. The confusion may lead to two problems. The first is duplicate predictions. Although DETR-like models can inhibit duplicate boxes with the help of set-based loss and self-attention [3], this ability is limited. As shown in the left figure of Fig. 8, when replacing our CDN queries with DN queries, the boy pointed by the arrow has 3 duplicate predictions. With CDN queries, our model can distinguish the slight difference between anchors and avoid duplicate predictions as shown in the right figure of Fig. 8. The second problem is that an unwanted anchor farther from a GT box might be selected. Although denoising training [17] has improved the model to choose nearby anchors, CDN further improves this capability by teaching the model to reject farther anchors.

Effectiveness: To demonstrate the effectiveness of CDN, we define a metric called Average Top-K Distance (ATD(k)) and use it to evaluate how far anchors are from their target GT boxes in the matching part. As in DETR, each anchor corresponds to a prediction which may be matched with a GT box or background. We only consider those matched with GT boxes here. Assume we have N GT bounding boxes $b_0, b_2, ..., b_{N-1}$ in a validation set, where $b_i = (x_i, y_i, w_i, h_i)$. For

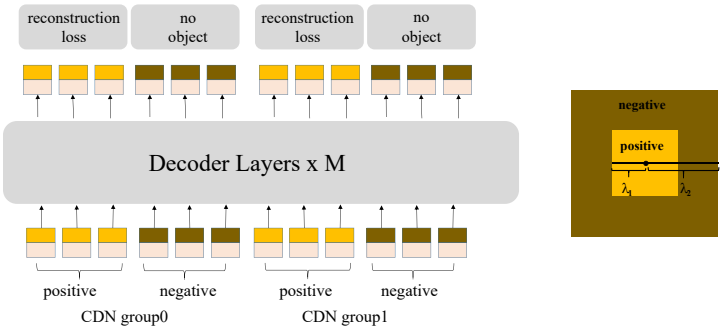

Fig. 3. CDN组的结构以及正负样本的展示。尽管正负样本均为可表示为4D空间点的4D锚点，但为简化起见，我们将其示意为同心正方形上的2D空间点。假设正方形中心为GT框，则内正方形内的点被视为正样本，内正方形与外正方形之间的点则视为负样本。

GT框对提升性能更有帮助。如图3所示，每个CDN组包含一组正查询和负查询。若图像含有n个GT框，则一个CDN组将生成$2\times n$个查询——每个GT框对应一个正查询和一个负查询。与DN-DETR类似，我们也采用多个CDN组来增强方法效果。重构损失包括用于边界框回归的l_1和GIOU损失，以及用于分类的focal loss[19]。将负样本分类为背景的损失同样采用focal loss。

Analysis: 我们的方法之所以有效，是因为它能够抑制混淆并为预测边界框筛选高质量锚点（查询）。当多个锚点靠近同一物体时，就会发生混淆现象。这种情况下，模型难以决定选择哪个锚点。这种混淆可能引发两个问题：首先是重复预测。尽管类DETR模型能借助基于集合的损失函数和自注意力机制[3]抑制重复框，但这种能力有限。如图8左图所示，当用DN查询替换我们的CDN查询时，箭头所指的男孩出现了3个重复预测框。而采用CDN查询后，我们的模型能辨识锚点间的细微差异，避免重复预测（见图8右图）。第二个问题是可能选中距离真实框较远的不理想锚点。虽然去噪训练[17]提升了模型选择邻近锚点的能力，但CDN通过教导模型拒绝较远锚点，进一步强化了这一能力。

Effectiveness: 为了验证CDN的有效性，我们定义了一个名为平均Top-K距离（ATD(k）的指标，并用它来评估在匹配部分中锚点与其目标真实框（GT框）之间的距离。与DETR类似，每个锚点对应一个预测，该预测可能与一个GT框或背景匹配。在此，我们仅考虑那些与GT框匹配的情况。假设在验证集中我们有N个GT边界框$b_0, b_2, ..., b_{N-1}$，其中$b_i = (x_i, y_i, w_i, h_i)$。对于

each b_i, we can find its corresponding anchor and denote it as $a_i = (x'_i, y'_i, w'_i, h'_i)$. a_i is the initial anchor box of the decoder whose refined box after the last decoder layer is assigned to b_i during matching. Then we have

$$ATD(k) = \frac{1}{k} \sum \{topK\left(\{\|b_0 - a_0\|_1, \|b_1 - a_1\|_1, ..., \|b_{N-1} - a_{N-1}\|_1\}, k\right)\} \quad (1)$$

where $\|b_i - a_i\|_1$ is the l_1 distance between b_i and a_i and $topK(\mathbf{x}, k)$ is a function that returns the set of k largest elements in $\mathbf{x}$. The reason why we select the top-K elements is that the confusion problem is more likely to happen when the GT box is matched with a farther anchor. As shown in (a) and (b) of Fig. 4, DN is good enough for selecting a good anchor overall. However, CDN finds better anchors for small objects. Fig. 4 (c) shows that CDN queries lead to an improvement of +1.3 AP over DN queries on small objects in 12 epochs with ResNet-50 and multi-scale features.

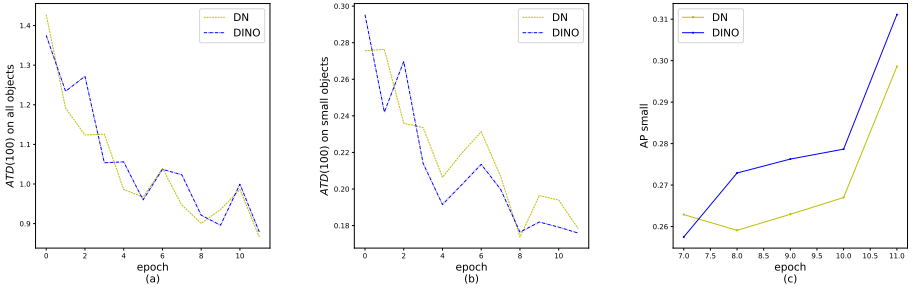

Fig. 4. (a) and (b) $ATD(100)$ on all objects and small objects respectively. (c) The AP on small objects.

3.4 Mixed Query Selection

In DETR [3] and DN-DETR [17], decoder queries are static embeddings without taking any encoder features from an individual image, as shown in Fig. 5 (a). They learn anchors (in DN-DETR and DAB-DETR) or positional queries (in DETR) from training data directly and set the content queries as all 0 vectors. Deformable DETR [41] learns both the positional and content queries, which is another implementation of static query initialization. To further improve the performance, Deformable DETR [41] has a query selection variant (called "two-stage" in [41]), which select top K encoder features from the last encoder layer as priors to enhance decoder queries. As shown in Fig. 5 (b), both the positional and content queries are generated by a linear transform of the selected features. In addition, these selected features are fed to an auxiliary detection head to get predicted boxes, which are used to initialize reference boxes. Similarly, Efficient DETR [39] also selects top K features based on the objectiveness (class) score of each encoder feature.

对于每个b_i，我们可以找到其对应的锚点并将其表示为$a_i = (x_i', y_i', w_i', h_i')$。$a_i$是解码器的初始锚框，其经过最后一层解码器细化后的框在匹配过程中被分配给b_i。于是我们有

$$ATD(k) = \frac{1}{k} \sum \{topK\left(\{\|b_0 - a_0\|_1, \|b_1 - a_1\|_1, ..., \|b_{N-1} - a_{N-1}\|_1\}, k\right)\{\mathrm{v}^{\mathrm{t}}\}$$

其中$\|b_i - a_i\|_1$表示b_i与a_i之间的l_1距离，$topK(\mathbf{x}, k)$是一个返回$\mathbf{x}$中k个最大元素的函数。我们选择前K个元素的原因是，当GT框与较远的锚点匹配时，更容易出现混淆问题。如图4(a)和(b)所示，DN在整体锚点选择上表现良好。然而，CDN能为小物体找到更优的锚点。图4(c)表明，在使用ResNet-50和多尺度特征的12个训练周期中，CDN查询相比DN查询在小物体上实现了+1.3 AP的性能提升。

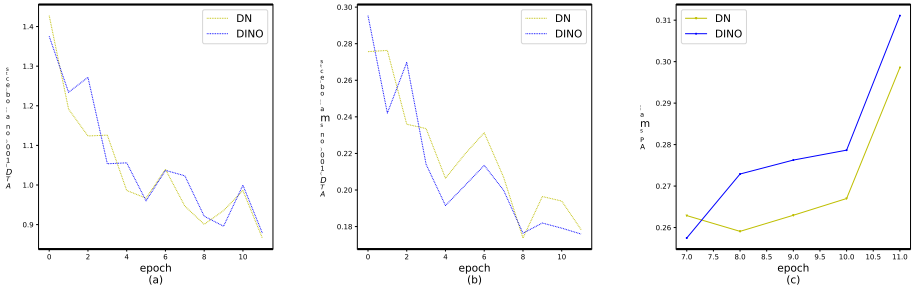

Fig. 4. (a)和(b)$ATD(100)$分别针对所有物体和小物体。(c)小物体上的AP。

3.4 Mixed Query Selection

在DETR [3]和DN-DETR [17]中，解码器查询是静态嵌入，不包含来自单个图像的编码器特征，如图5(a)所示。它们直接从训练数据中学习锚点（在DN-DETR和DAB-DETR中）或位置查询（在DETR中），并将内容查询设置为全0向量。可变形DETR [41]则同时学习位置查询和内容查询，这是静态查询初始化的另一种实现方式。为了进一步提升性能，可变形DETR [41]提出了一个查询选择变体（在[41]中称为“两阶段”），该变体从最后一个编码器层选取前K个编码器特征作为先验信息来增强解码器查询。如图5(b)所示，位置查询和内容查询均通过对所选特征的线性变换生成。此外，这些选中的特征会被送入一个辅助检测头以获取预测框，用于初始化参考框。类似地，高效DETR [39]也基于每个编码器特征的目标性（类别）分数选取前K个特征。

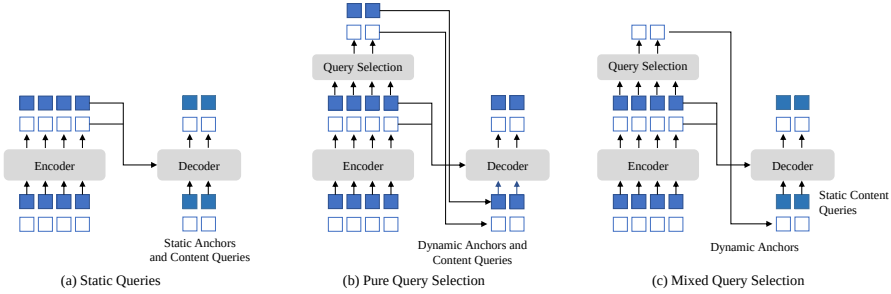

Fig. 5. Comparison of three different query initialization methods. The term "static" means that they will keep the same for different images in inference. A common implementation for these static queries is to make them learnable.

The dynamic 4D anchor box formulation of queries in our model makes it closely related to decoder positional queries, which can be improved by query selection. We follow the above practice and propose a mixed query selection approach. As shown in Fig. 5 (c), we only initialize anchor boxes using the position information associated with the selected top-K features, but leave the content queries static as before. Note that Deformable DETR [41] utilizes the top-K features to enhance not only the positional queries but also the content queries. As the selected features are preliminary content features without further refinement, they could be ambiguous and misleading to the decoder. For example, a selected feature may contain multiple objects or be only part of an object. In contrast, our mixed query selection approach only enhances the positional queries with top-K selected features and keeps the content queries learnable as before. It helps the model to use better positional information to pool more comprehensive content features from the encoder.

3.5 Look Forward Twice

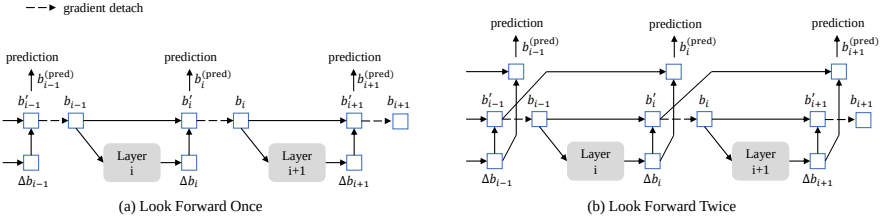

Fig. 6. Comparison of box update in Deformable DETR and our method.

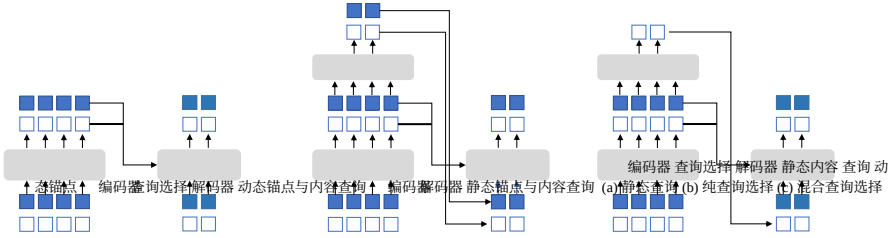

Fig. 5. 三种不同查询初始化方法的比较。术语“静态”意味着在推理过程中，它们对于不同图像保持不变。这些静态查询的一种常见实现方式是使其可学习。

我们模型中动态4D锚框查询的构建方式使其与解码器的位置查询密切相关，后者可通过查询选择进行优化。我们遵循上述实践，提出了一种混合查询选择方法。如图5(c)所示，我们仅利用所选Top-K特征关联的位置信息初始化锚框，而保持内容查询与先前一致静态不变。值得注意的是，可变形DETR[41]不仅利用Top-K特征增强位置查询，还改进了内容查询。由于所选特征属于未经进一步优化的初步内容特征，可能具有模糊性并对解码器产生误导——例如某个选定特征可能包含多个物体或仅为物体局部。相比之下，我们的混合查询选择方法仅通过Top-K选定特征增强位置查询，同时保持内容查询与先前相同的可学习性。这有助于模型利用更精准的位置信息，从编码器中汇聚更全面的内容特征。

3.5 Look Forward Twice

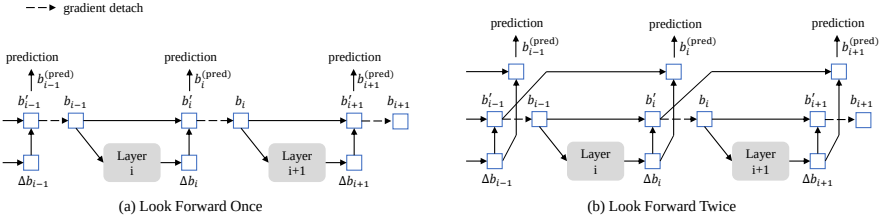

Fig. 6. 可变形DETR与我们方法中边界框更新的对比。

We propose a new way to box prediction in this section. The iterative box refinement in Deformable DETR [41] blocks gradient back propagation to stabilize training. We name the method look forward once since the parameters of layer i are updated based on the auxiliary loss of boxes b_i only, as shown in Fig. 6 (a). However, we conjecture that the improved box information from a later layer could be more helpful to correct the box prediction in its adjacent early layer. Hence we propose another way called look forward twice to perform box update, where the parameters of layer-i are influenced by losses of both layer-i and layer-$(i+1)$, as shown in Fig. 6 (b). For each predicted offset Δb_i, it will be used to update box twice, one for b'_i and another for $b_{i+1}^{(pred)}$, hence we name our method as look forward twice.

The final precision of a predicted box $b_i^{(\text{pred})}$ is determined by two factors: the quality of the initial box b_{i-1} and the predicted offset of the box Δb_i. The look forward once scheme optimizes the latter only, as the gradient information is detached from layer-i to layer-$(i-1)$. In contrast, we improve both the initial box b_{i-1} and the predicted box offset Δb_i. A simple way to improving the quality is supervising the final box b'_i of layer i with the output of the next layer Δb_{i+1}. Hence we use the sum of b'_i and Δb_{i+1} as the predicted box of layer-$(i+1)$.

More specifically, given an input box b_{i-1} for the i-th layer, we obtain the final prediction box $b_i^{(\text{pred})}$ by:

$$\begin{aligned} \Delta b_i &= \text{Layer}_\text{i}(b_{i-1}), & b'_i &= \text{Update}(b_{i-1}, \Delta b_i), \\ b_i &= \text{Detach}(b'_i), & b_i^{(\text{pred})} &= \text{Update}(b'_{i-1}, \Delta b_i), \end{aligned} \tag{2}$$

where b'_i is the undetached version of b_i. The term $\text{Update}(\cdot, \cdot)$ is a function that refines the box b_{i-1} by the predicted box offset Δb_i. We adopt the same way for box update[3] as in Deformable DETR [41].

4 Experiments

4.1 Setup

Dataset and Backbone: We conduct evaluation on the COCO 2017 object detection dataset [20], which is split into `train2017` and `val2017` (also called `minival`). We report results with two different backbones: ResNet-50 [14] pre-trained on ImageNet-1k [9] and SwinL [23] pre-trained on ImageNet-22k [9]. DINO with ResNet-50 is trained on `train2017` without extra data, while DINO with SwinL is first pre-trained on Object365 [33] and then fine-tuned on `train2017`. We report the standard average precision (AP) result on `val2017` under different IoU thresholds and object scales. We also report the `test-dev` results for DINO with SwinL.

[3] We use normalized forms of boxes in our model, hence each value of a box is a float between 0 and 1. Given two boxes, we sum them after inverse sigmoid and then transform the summation by sigmoid. Refer to Deformable DETR [41] Sec. A.3 for more details.

在本节中，我们提出了一种新的边界框预测方法。可变形DETR[41]中的迭代框优化通过阻断梯度反向传播来稳定训练。我们将该方法命名为"前瞻一次"，因为如图6(a)所示，层i的参数仅基于边界框b_i的辅助损失进行更新。但我们推测，来自更深层的优化框信息可能更有助于修正其相邻浅层的框预测。因此，我们提出了另一种称为"前瞻两次"的框更新方式，其中层i的参数同时受到层i和层$i+1$损失的共同影响，如图6(b)所示。每个预测偏移量Δb_i将被用于两次框更新——分别作用于b'_i和$b_{i+1}^{(pred)}$，故将此方法命名为"前瞻两次"。

预测框$b_i^{(\text{pred})}$的最终精度由两个因素决定：初始框b_{i-1}的质量和该框Δb_i的预测偏移量。前瞻一次方案仅优化后者，因为梯度信息从第i层到第(i -1)层被截断。相比之下，我们同时改进了初始框b_{i-1}和预测框偏移量Δb_i。提升质量的一个简单方法是用下一层Δb_{i+1}的输出监督第i层的最终框b'_i。因此，我们采用b'_i与Δb_{i+1}之和作为第(i +1)层的预测框。

更具体地说，给定第i层的输入框b_{i-1}，我们通过以下方式得到最终的预测框$b_i^{(\text{pred})}$：

$$\begin{aligned} \Delta b_i &= \text{Layer}_\text{i}(b_{i-1}), & b'_i &= \text{Update}(b_{i-1}, \Delta b_i), \\ b_i &= \text{Detach}(b'_i), & b_i^{(\text{pred})} &= \text{Update}(b'_{i-1}, \Delta b_i), \end{aligned} \tag{2}$$

其中b'_i是b_i的未分离版本。术语Update$(\cdot,\cdot)$是一个函数，通过预测的边界框偏移Δb_i来优化边界框b_{i-1}。我们采用与可变形DETR[41]中相同的边界框更新[3]方式。

4 Experiments

4.1 Setup

Dataset and Backbone: 我们在COCO 2017目标检测数据集[20]上进行评估，该数据集分为train2017和val2017（亦称minival）。我们采用两种不同骨干网络报告结果：基于ImageNet-1k[9]预训练的ResNet-50[14]和基于ImageNet-22k[9]预训练的SwinL[23]。使用ResNet-50的DINO仅在train2017上训练，未使用额外数据；而采用SwinL的DINO则先在Object365[33]上预训练，再于train2017上微调。我们报告了val2017在不同IoU阈值和物体尺度下的标准平均精度（AP）结果，同时提供了SwinL版DINO在test-dev集上的表现。

3 我们在模型中使用了归一化的边界框形式，因此每个边界框的值都是介于0到1之间的浮点数。给定两个边界框，我们先对它们进行逆sigmoid变换后求和，再通过sigmoid函数对求和结果进行转换。更多细节请参考Deformable DETR [41] 章节A.3。

Implementation Details: DINO is composed of a backbone, a Transformer encoder, a Transformer decoder, and multiple prediction heads. In appendix D, we provide more implementation details, including all the hyper-parameters and engineering techniques used in our models for those who want to reproduce our results. We will release the code after the blind review.

4.2 Main Results

Model	Epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	GFLOPS	Params	FPS
Faster-RCNN(5scale) [30]	12	37.9	58.8	41.1	22.4	41.1	49.1	207	40M	21*
DETR(DC5) [3]	12	15.5	29.4	14.5	4.3	15.1	26.7	225	41M	20
Deformable DETR(4scale)[41]	12	41.1	–	–	–	–		196	40M	24
DAB-DETR(DC5)† [21]	12	38.0	60.3	39.8	19.2	40.9	55.4	256	44M	17
Dynamic DETR(5scale) [8]	12	42.9	61.0	46.3	24.6	44.9	54.4	–	58M	–
Dynamic Head(5scale) [7]	12	43.0	60.7	46.8	24.7	46.4	53.9	–	–	–
HTC(5scale) [4]	12	42.3	–	–	–	–	–	441	80M	5*
DN-Deformable-DETR(4scale)† [17]	12	43.4	61.9	47.2	24.8	46.8	59.4	265	48M	23
DINO-4scale†	12	**49.0**(+5.6)	**66.6**	**53.5**	**32.0**(+7.2)	**52.3**	**63.0**	279	47M	24
DINO-5scale†	12	**49.4**(+6.0)	**66.9**	**53.8**	**32.3**(+7.5)	**52.5**	**63.9**	860	47M	10

Table 1. Results for DINO and other detection models with the ResNet50 backbone on COCO `val2017` trained with 12 epochs (the so called 1× setting). For models without multi-scale features, we test their GFLOPS and FPS for their best model ResNet-50-DC5. DINO uses 900 queries. † indicates models that use 900 queries or 300 queries with 3 patterns which has similar effect with 900 queries. Other DETR-like models except DETR (100 queries) uses 300 queries. * indicates that they are tested using the mmdetection [5] framework.

12-epoch setting: With our improved anchor box denoising and training losses, the training process can be significantly accelerated. As shown in Table 1, we compare our method with strong baselines including both convolution-based methods [30,4,7] and DETR-like methods [3,41,8,21,17]. For a fair comparison, we report both GFLOPS and FPS tested on the same A100 NVIDIA GPU for all the models listed in Table 1. All methods except for DETR and DAB-DETR use multi-scale features. For those without multi-scale features, we report their results with ResNet-DC5 which has a better performance for its use of a dilated larger resolution feature map. Since some methods adopt 5 scales of feature maps and some adopt 4, we report our results with both 4 and 5 scales of feature maps.

As shown in Table 1, our method yields an improvement of +5.6 AP under the same setting using ResNet-50 with 4-scale feature maps and +6.0 AP with 5-scale feature maps. Our 4-scale model does not introduce much overhead in computation and the number of parameters. Moreover, our method performs especially well for small objects, gaining +7.2 AP with 4 scales and +7.5 AP with 5 scales. Note that the results of our models with ResNet-50 backbone are higher than those in the first version of our paper due to engineering techniques.

Comparison with the best models with a ResNet-50 backbone: To

Implementation Details: DINO由一个主干网络、一个Transformer编码器、一个Transformer解码器以及多个预测头组成。在附录D中，我们为希望复现我们成果的研究者提供了更多实现细节，包括模型中使用的所有超参数和工程技巧。代码将在盲审结束后公开。

4.2 Main Results

Model	Epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	GFLOPS	Params	FPS
Faster-RCNN(5scale) [30]	12	37.9	58.8	41.1	22.4	41.1	49.1	207	40M	21*
DETR(DC5) [3]	12	15.5	29.4	14.5	4.3	15.1	26.7	225	41M	20
Deformable DETR(4scale)[41]	12	41.1	–	–	–	–		196	40M	24
DAB-DETR(DC5)† [21]	12	38.0	60.3	39.8	19.2	40.9	55.4	256	44M	17
Dynamic DETR(5scale) [8]	12	42.9	61.0	46.3	24.6	44.9	54.4	–	58M	–
Dynamic Head(5scale) [7]	12	43.0	60.7	46.8	24.7	46.4	53.9	–	–	–
HTC(5scale) [4]	12	42.3	–	–	–	–	–	441	80M	5*
DN-Deformable-DETR(4scale)† [17]	12	43.4	61.9	47.2	24.8	46.8	59.4	265	48M	23
DINO-4scale†	12	**49.0**(+5.6)	**66.6**	**53.5**	**32.0**(+7.2)	**52.3**	**63.0**	279	47M	24
DINO-5scale†	12	**49.4**(+6.0)	**66.9**	**53.8**	**32.3**(+7.5)	**52.5**	**63.9**	860	47M	10

Table 1. 在COCO val2017数据集上，采用ResNet50骨干网络并训练12个周期（即所谓的1×设置）的DINO及其他检测模型的结果。对于不具备多尺度特征的模型，我们测试了其最佳模型ResNet-50-DC5的GFLOPS和FPS性能。DINO使用了900个查询。†标记表示采用900个查询或具有相似效果的3种模式300个查询的模型。除DETR（使用100个查询）外，其他类DETR模型均采用300个查询。*标记表示这些模型是通过mmdetection[5]框架进行测试的。

12-epoch setting: 通过我们改进的锚框去噪和训练损失，训练过程可以显著加速。如表1所示，我们将本方法与包括基于卷积的方法[30,4,7]和类DETR方法[3,41,8,21,17]在内的强基线进行了比较。为确保公平对比，我们报告了所有表1列出的模型在相同A100 NVIDIA GPU上测试的GFLOPS和FPS指标。除DETR和DAB-DETR外，所有方法均采用多尺度特征。对于不使用多尺度特征的方法，我们报告其采用ResNet-DC5（通过扩张更大分辨率特征图获得更好性能）的结果。由于部分方法采用5层特征图尺度而部分采用4层，我们分别报告了本方法在4层和5层特征图尺度下的结果。

如表1所示，在相同设置下，我们的方法采用ResNet-50骨干网络配合4尺度特征图时实现了+5.6 AP的提升，而使用5尺度特征图时则达到+6.0 AP。我们的4尺度模型在计算量和参数量上并未引入过多开销。此外，本方法对小目标检测表现尤为突出，4尺度下获得+7.2 AP，5尺度下取得+7.5 AP。需注意的是，由于工程优化技术的应用，基于ResNet-50的模型性能较论文初版有所提升。

Comparison with the best models with a ResNet-50 backbone:

Model	Epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster-RCNN [30]	108	42.0	62.4	44.2	20.5	45.8	61.1
DETR(DC5) [41]	500	43.3	63.1	45.9	22.5	47.3	61.1
Deformable DETR [41]	50	46.2	65.2	50.0	28.8	49.2	61.7
SMCA-R [11]	50	43.7	63.6	47.2	24.2	47.0	60.4
TSP-RCNN-R [34]	96	45.0	64.5	49.6	29.7	47.7	58.0
Dynamic DETR(5scale) [7]	50	47.2	65.9	51.1	28.6	49.3	59.1
DAB-Deformable-DETR [21]	50	46.9	66.0	50.8	30.1	50.4	62.5
DN-Deformable-DETR [17]	50	48.6	67.4	52.7	31.0	52.0	63.7
DINO-4scale	24	**50.4**(+1.8)	68.3	54.8	33.3	53.7	64.8
DINO-5scale	24	**51.3**(+2.7)	69.1	56.0	34.5	54.2	65.8
DINO-4scale	36	**50.9**(+2.3)	69.0	55.3	34.6	54.1	64.6
DINO-5scale	36	**51.2**(+2.6)	69.0	55.8	35.0	54.3	65.3

Table 2. Results for DINO and other detection models with the ResNet-50 backbone on COCO `val2017` trained with more epochs (24, 36, or more).

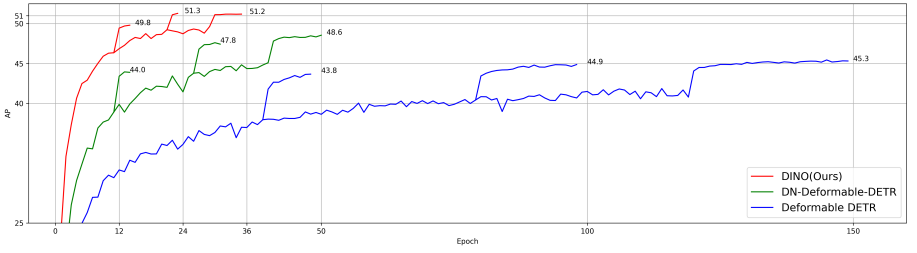

Fig. 7. Training convergence curves evaluated on COCO `val2017` for DINO and two previous state-of-the-art models with ResNet-50 using multi-scale features.

validate the effectiveness of our method in improving both convergence speed and performance, we compare our method with several strong baselines using the same ResNet-50 backbone. Despite the most common 50-epoch setting, we adopt the 24 (2×) and 36 (3×) epoch settings since our method converges faster and yields only a smaller additional gain with 50-epoch training. The results in Table 2 show that, using only 24 epochs, our method achieves an improvement of +1.8 AP and +2.7 AP with 4 and 5 scales, respectively. Moreover, using 36 epochs in the 3× setting, the improvement increases to +2.3 and +2.6 AP with 4 and 5 scales, respectively. The detailed convergence curve comparison is shown in Fig. 7.

Model	Epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster-RCNN [30]	108	42.0	62.4	44.2	20.5	45.8	61.1
DETR(DC5) [41]	500	43.3	63.1	45.9	22.5	47.3	61.1
Deformable DETR [41]	50	46.2	65.2	50.0	28.8	49.2	61.7
SMCA-R [11]	50	43.7	63.6	47.2	24.2	47.0	60.4
TSP-RCNN-R [34]	96	45.0	64.5	49.6	29.7	47.7	58.0
Dynamic DETR(5scale) [7]	50	47.2	65.9	51.1	28.6	49.3	59.1
DAB-Deformable-DETR [21]	50	46.9	66.0	50.8	30.1	50.4	62.5
DN-Deformable-DETR [17]	50	48.6	67.4	52.7	31.0	52.0	63.7
DINO-4scale	24	**50.4**(+1.8)	68.3	54.8	33.3	53.7	64.8
DINO-5scale	24	**51.3**(+2.7)	69.1	56.0	34.5	54.2	65.8
DINO-4scale	36	**50.9**(+2.3)	69.0	55.3	34.6	54.1	64.6
DINO-5scale	36	**51.2**(+2.6)	69.0	55.8	35.0	54.3	65.3

Table 2. 在COCO val2017数据集上，采用ResNet-50骨干网络并训练更多周期（24、36或更多）的DINO及其他检测模型的结果。

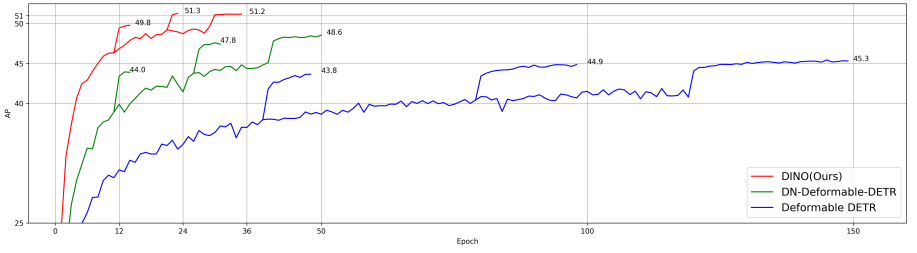

Fig. 7. 在COCO val2017数据集上评估的DINO与两种先前最先进模型（采用ResNet-50及多尺度特征）的训练收敛曲线。

为验证我们方法在提升收敛速度与性能上的有效性，我们采用相同ResNet-50主干网络与多个强基线方法进行对比。尽管最常见的训练周期为50轮，但由于我们的方法收敛更快且在50轮训练中仅能获得较小增益，因此采用24轮（2×）和36轮（3×）的设置。表2结果显示：在仅使用24轮训练时，我们的方法在4尺度与5尺度下分别实现了+1.8 AP与+2.7 AP的提升；而在3×设置的36轮训练中，提升幅度进一步扩大至4尺度+2.3 AP与5尺度+2.6 AP。具体收敛曲线对比见图7。

Method	Params	Backbone Pre-training Dataset	Detection Pre-training Dataset	Use Mask	End-to-end	val2017 (AP)		test-dev (AP)	
						w/o TTA	w/ TTA	w/o TTA	w/ TTA
SwinL [23]	284M	IN-22K-14M	O365	✓		57.1	58.0	57.7	58.7
DyHead [7]	≥ 284M	IN-22K-14M	Unknown*			–	58.4	–	60.6
Soft Teacher+SwinL [38]	284M	IN-22K-14M	O365	✓		60.1	60.7	–	61.3
GLIP [18]	≥ 284M	IN-22K-14M	FourODs [18],GoldG+ [18,15]			–	60.8	–	61.5
Florence-CoSwin-H[40]	≥ 637M	FLD-900M [40]	FLD-9M [40]			–	62.0	–	62.4
SwinV2-G [22]	3.0B	IN-22K-ext-70M [22]	O365	✓		61.9	62.5	–	63.1
DINO-SwinL(Ours)	**218M**	IN-22K-14M	O365		✓	**63.1**	**63.2**	**63.2**	**63.3**

Table 3. Comparison of the best detection models on MS-COCO. Similar to DETR [3], we use the term "end-to-end" to indicate if a model is free from hand-crafted components like RPN and NMS. The term "use mask" means whether a model is trained with instance segmentation annotations. We use the terms "IN" and "O365" to denote the ImageNet [9] and Objects365 [33] datasets, respectively. Note that "O365" is a subset of "FourODs" and "FLD-9M". * DyHead does not disclose the details of the datasets used for model pre-training.

4.3 Comparison with SOTA Models

To compare with SOTA results, we use the publicly available SwinL [23] backbone pre-trained on ImageNet-22K. We first pre-train DINO on the Objects365 [33] dataset and then fine-tune it on COCO. As shown in Table 3, DINO achieves the best results of 63.2AP and 63.3AP on COCO `val2017` and `test-dev`, which demonstrate its strong scalability to larger model size and data size. Note that all the previous SOTA models in Table 3 do not use Transformer decoder-based detection heads (HTC++ [4] and DyHead [7]). It is the first time that an end-to-end Transformer detector is established as a SOTA model on the leaderboard [1]. Compared with the previous SOTA models, we use a much smaller model size (1/15 parameters compared with SwinV2-G [22]), backbone pre-training data size (1/60 images compared with Florence), and detection pre-training data size (1/5 images compared with Florence), while achieving better results. In addition, our reported performance without test time augmentation (TTA) is a neat result without bells and whistles. These results effectively show the superior detection performance of DINO compared with traditional detectors.

4.4 Ablation

Effectiveness of New Algorithm Components: To validate the effectiveness of our proposed methods, we build a strong baseline with optimized DN-DETR and pure query selection as described in section 3.1. We include all the pipeline optimization and engineering techniques (see section 4.1 and Appendix D) in the strong baseline. The result of the strong baseline is available in Table 4 Row 3. We also present the result of optimized DN-DETR without pure query selection from Deformable DETR [41] in Table 4 Row 2. While our strong baseline outperforms all previous models, our three new methods in DINO further improve the performance significantly.

Method	Params	Backbone Pre-training Dataset	Detection Pre-training Dataset	Use Mask	End-to-end	val2017 (AP)		test-dev (AP)	
						w/o TTA	w/ TTA	w/o TTA	w/ TTA
SwinL [23]	284M	IN-22K-14M	O365	✓		57.1	58.0	57.7	58.7
DyHead [7]	≥ 284M	IN-22K-14M	Unknown*			–	58.4	–	60.6
Soft Teacher+SwinL [38]	284M	IN-22K-14M	O365	✓		60.1	60.7	–	61.3
GLIP [18]	≥ 284M	IN-22K-14M	FourODs [18],GoldG+ [18,15]			–	60.8	–	61.5
Florence-CoSwin-H[40]	≥ 637M	FLD-900M [40]	FLD-9M [40]			–	62.0	–	62.4
SwinV2-G [22]	3.0B	IN-22K-ext-70M [22]	O365	✓		61.9	62.5	–	63.1
DINO-SwinL(Ours)	**218M**	IN-22K-14M	O365		✓	**63.1**	**63.2**	**63.2**	**63.3**

Table 3. MS-COCO上最佳检测模型的对比。与DETR[3]类似，我们使用“端到端”这一术语来表示模型是否避免了手工设计的组件，如RPN和NMS。“使用掩码”指的是模型是否使用了实例分割标注进行训练。我们用“IN”和“O365”分别表示ImageNet[9]和Objects365[33]数据集。请注意，“O365”是“FourODs”和“FLD-9M”的一个子集。*DyHead未公开用于模型预训练的数据集细节。

4.3 Comparison with SOTA Models

为与SOTA结果进行比较，我们采用公开可用的SwinL[23]骨干网络，该网络在ImageNet-22K上进行了预训练。我们首先在Objects365[33]数据集上对DINO进行预训练，随后在COCO上进行微调。如表3所示，DINO在COCO val2017和test-dev上分别取得了63.2AP和63.3AP的最佳成绩，这证明了其在更大模型规模和数据规模下的强大扩展能力。值得注意的是，表3中所有先前的SOTA模型均未使用基于Transformer解码器的检测头（如HTC++[4]和DyHead[7]）。这是首次有端到端Transformer检测器在排行榜[1]上确立为SOTA模型。与先前SOTA模型相比，我们使用的模型规模更小（参数数量为1/15，而SwinV2-G[22]更多）、骨干网络预训练数据量更少（1/60张图像对比Florence）、检测预训练数据量更少（1/5张图像对比Florence），却实现了更优的结果。此外，我们报告的性能未使用测试时增强（TTA），是未经修饰的纯粹结果。这些结果有效证明了DINO相较于传统检测器的卓越检测性能。

4.4 Ablation

Effectiveness of New Algorithm Components: 为验证我们提出方法的有效性，我们基于3.1节描述的优化版DN-DETR与纯查询选择构建了一个强基线模型。该强基线整合了全部流程优化与工程实现技术（详见4.1节与附录D）。强基线的实验结果展示在表4第3行。我们同时在表4第2行列出了基于可变形DETR[41]、未使用纯查询选择的优化版DN-DETR结果。虽然我们的强基线已超越所有先前模型，但DINO中提出的三项新方法仍能显著提升性能。

#Row	QS	CDN	LFT	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
1. DN-DETR [17]	No			43.4	61.9	47.2	24.8	46.8	59.4
2. Optimized DN-DETR	No			44.9	62.8	48.6	26.9	48.2	60.0
3. Strong baseline (Row2+pure query selection)	Pure			46.5	64.2	50.4	29.6	49.8	61.0
4. Row3+mixed query selection	Mixed			47.0	64.2	51.0	31.1	50.1	61.5
5. Row4+look forward twice	Mixed		✓	47.4	64.8	51.6	29.9	50.8	61.9
6. DINO (ours, Row5+contrastive DN)	Mixed	✓	✓	**47.9**	**65.3**	**52.1**	**31.2**	**50.9**	**61.9**

Table 4. Ablation comparison of the proposed algorithm components. We use the terms "QS", "CDN", and "LFT" to denote "Query Selection", "Contrastive De-Noising Training", and "Look Forward Twice", respectively.

5 Conclusion

In this paper, we have presented a strong end-to-end Transformer detector DINO with contrastive denoising training, mixed query selection, and look forward twice, which significantly improves both the training efficiency and the final detection performance. As a result, DINO outperforms all previous ResNet-50-based models on COCO `val2017` in both the 12-epoch and the 36-epoch settings using multi-scale features. Motivated by the improvement, we further explored to train DINO with a stronger backbone on a larger dataset and achieved a new state of the art, 63.3 AP on COCO 2017 `test-dev`. This result establishes DETR-like models as a mainstream detection framework, not only for its novel end-to-end detection optimization, but also for its superior performance.

Fig. 8. The left figure is the detection result using a model trained with DN queries and the right is the result of our method. In the left image, the boy pointed by the arrow has 3 duplicate bounding boxes. For clarity, we only show boxes of class "person".

#Row	QS	CDN	LFT	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
1. DN-DETR [17]	No			43.4	61.9	47.2	24.8	46.8	59.4
2. Optimized DN-DETR	No			44.9	62.8	48.6	26.9	48.2	60.0
3. Strong baseline (Row2+pure query selection)	Pure			46.5	64.2	50.4	29.6	49.8	61.0
4. Row3+mixed query selection	Mixed			47.0	64.2	51.0	31.1	50.1	61.5
5. Row4+look forward twice	Mixed		✓	47.4	64.8	51.6	29.9	50.8	61.9
6. DINO (ours, Row5+contrastive DN)	Mixed	✓	✓	**47.9**	**65.3**	**52.1**	**31.2**	**50.9**	**61.9**

Table 4. 所提算法组件的消融对比。我们分别用“QS”、“CDN”和“LFT”来指代“查询选择”、“对比去噪训练”和“前瞻两次”。

5 Conclusion

本文提出了一种强大的端到端Transformer检测器DINO，它结合了对比去噪训练、混合查询选择及前瞻两次策略，显著提升了训练效率和最终检测性能。实验表明，在COCO val2017数据集上，无论是12周期还是36周期多尺度特征训练，DINO均超越了所有基于ResNet-50的现有模型。基于这一改进，我们进一步探索采用更强主干网络在更大规模数据集上训练DINO，最终在COCO 2017 test-dev上以63.3 AP的成绩刷新了当前最优水平。这一成果确立了类DETR模型作为主流检测框架的地位，不仅因其新颖的端到端检测优化机制，更因其卓越的性能表现。

Fig. 8. 左图展示了使用DN查询训练的模型检测结果，右图则是我们方法的效果。在左侧图像中，箭头所指的男孩存在3个重复的边界框。为清晰起见，我们仅显示类别为“人”的检测框。

References

1. Papers with code - coco test-dev benchmark (object detection).
2. Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.
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.
4. Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4974–4983, 2019.
5. Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.
6. Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.
7. Xiyang Dai, Yinpeng Chen, Bin Xiao, Dongdong Chen, Mengchen Liu, Lu Yuan, and Lei Zhang. Dynamic head: Unifying object detection heads with attentions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7373–7382, 2021.
8. Xiyang Dai, Yinpeng Chen, Jianwei Yang, Pengchuan Zhang, Lu Yuan, and Lei Zhang. Dynamic detr: End-to-end object detection with dynamic attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2988–2997, October 2021.
9. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
10. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
11. Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fast convergence of detr with spatially modulated co-attention. *arXiv preprint arXiv:2101.07448*, 2021.
12. Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.
13. Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.
14. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
15. Aishwarya Kamath, Mannat Singh, Yann LeCun, Ishan Misra, Gabriel Synnaeve, and Nicolas Carion. Mdetr – modulated detection for end-to-end multi-modal understanding. *arXiv: Computer Vision and Pattern Recognition*, 2021.
16. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
17. Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. *arXiv preprint arXiv:2203.01305*, 2022.

References

1. 论文与代码 - COCO test-dev基准测试（目标检测）。 2. Alexey Bochkovskiy、Chien-Yao Wang与Hong-Yuan Mark Liao。YOLOv4：目标检测的最佳速度与精度。*arXiv preprint arXiv:2004.10934*，2020年。 3. Nicolas Carion、Francisco Massa、Gabriel Synnaeve、Nicolas Usunier、Alexander Kirillov与Sergey Zagoruyko。基于Transformer的端到端目标检测。载于*European conference on computer vision*，第213–229页。Springer，2020年。 4. 陈凯、庞江淼、王佳琪、熊宇、李晓晓、孙书阳、冯万森、刘子纬、史建波、欧阳万里等。面向实例分割的混合任务级联。载于*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*，第4974–4983页，2019年。 5. 陈凯、王佳琪、庞江淼、曹宇航、熊宇、李晓晓、孙书阳、冯万森、刘子纬、徐佳瑞等。MMDetection：Open MMLab检测工具箱与基准。*arXiv preprint arXiv:1906.07155*，2019年。 6. 陈天奇、徐冰、张驰原与Carlos Guestrin。以次线性内存成本训练深度网络。*arXiv preprint arXiv:1604.06174*，2016年。 7. 戴曦阳、陈寅鹏、肖斌、陈栋栋、刘梦辰、苑露与张磊。动态头：通过注意力统一目标检测头。载于*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*，第7373–7382页，2021年。 8. 戴曦阳、陈寅鹏、杨建伟、张鹏川、苑露与张磊。动态DETR：基于动态注意力的端到端目标检测。载于*Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*，第2988–2997页，2021年10月。 9. 邓嘉、董伟、Richard Socher、李立佳、李凯与李飞飞。ImageNet：大规模分层图像数据库。载于*2009 IEEE conference on computer vision and pattern recognition*，第248–255页。IEEE，2009年。 10. Jacob Devlin、张明伟、Kenton Lee与Kristina Toutanova。BERT：面向语言理解的深度双向Transformer预训练。*arXivpreprint arXiv:1810.04805*，2018年。 11. 高鹏、郑明航、王晓刚、代继峰与李宏生。通过空间调制协同注意力实现DETR快速收敛。*arXiv preprint arXiv:2101.07448*，2021年。 12. 葛政、刘松涛、王峰、李泽明与孙剑。YOLOX：2021年超越YOLO系列。*arXiv preprint arXiv:2107.08430*，2021年。 13. 何恺明、Georgia Gkioxari、Piotr Dollár与Ross Girshick。Mask R-CNN。载于*Proceedings of the IEEE international conference on computer vision*，第2961–2969页，2017年。 14. 何恺明、张祥雨、任少卿与孙剑。深度残差学习用于图像识别。载于*2016 IEEE Conference on Computer Vision and PatternRecognition (CVPR)*，第770–778页，2016年。 15. Aishwarya Kamath、Mannat Singh、Yann LeCun、Ishan Misra、Gabriel Synnaeve与Nicolas Carion。MDETR——面向端到端多模态理解的调制检测。*arXiv: Computer Vision and Pattern Recognition*，2021年。 16. Diederik P Kingma与Jimmy Ba。Adam：一种随机优化方法。*arXiv preprint arXiv:1412.6980*，2014年。 17. 李锋、张浩、刘世龙、郭健、倪明选与张磊。DN-DETR：通过查询去噪加速DETR训练。*arXiv preprintarXiv:2203.01305*，2022年。

18. Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. *arXiv preprint arXiv:2112.03857*, 2021.
19. Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327, 2020.
20. 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 *European conference on computer vision*, pages 740–755. Springer, 2014.
21. 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.
22. Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. *arXiv preprint arXiv:2111.09883*, 2021.
23. Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.
24. Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
25. Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. *arXiv preprint arXiv:2108.06152*, 2021.
26. Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. Mixed precision training, 2018.
27. Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.
28. Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.
29. Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.
30. 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.
31. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.
32. Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 658–666, 2019.
33. 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.

18. 刘念Harold Li、张鹏川、张昊天、杨建伟、李春元、钟毅武、王丽娟、袁璐、张磊、黄正能等。基于地面的语言-图像预训练。*arXiv preprint arXiv:2112.03857*，2021年。 19. 林钲贻、Priya Goyal、Ross Girshick、何恺明、Piotr Dollar。密集目标检测的焦点损失。*IEEE Transactions on Pattern Analysis and MachineIntelligence*，42(2):318–327，2020年。 20. 林钲贻、Michael Maire、Serge Belongie、James Hays、Pietro Perona、Deva Ramanan、Piotr Dollár、C Lawrence Zitnick。Microsoft COCO：上下文中的常见物体。载于*European conference on computer vision*，第740–755页。Springer，2014年。 21. 刘世龙、李峰、张浩、杨晓、齐先彪、苏航、朱军、张磊。DAB-DETR：动态锚框是DETR更优的查询方式。*arXiv preprint arXiv:2201.12329*，2022年。 22. 刘泽、胡涵、林雨桐、姚竹亮、谢振达、魏亦轩、宁佳、曹越、张政、董力等。Swin Transformer V2：扩展容量与分辨率。*arXiv preprint arXiv:2111.09883*，2021年。 23. 刘泽、林雨桐、曹越、胡涵、魏亦轩、张政、Stephen Lin、郭百宁。Swin Transformer：基于移位窗口的分层视觉Transformer。载于*Proceedings of the IEEE/CVF International Conference on ComputerVision*，第10012–10022页，2021年。 24. Ilya Loshchilov、Frank Hutter。解耦权重衰减正则化。*arXiv preprint arXiv:1711.05101*，2017年。 25. 孟德普、陈晓康、范泽佳、曾刚、李厚强、袁钰辉、孙磊、王井东。条件式DETR实现快速训练收敛。*arXiv preprint arXiv:2108.06152*，2021年。 26. Paulius Micikevicius、Sharan Narang、Jonah Alben、Gregory Diamos、Erich Elsen、David Garcia、Boris Ginsburg、Michael Houston、Oleksii Kuchaiev、Ganesh Venkatesh、吴昊。混合精度训练，2018年。 27. Alec Radford、Jong Wook Kim、Chris Hallacy、Aditya Ramesh、Gabriel Goh、Sandhini Agarwal、Girish Sastry、Amanda Askell、Pamela Mishkin、Jack Clark、Gretchen Krueger、Ilya Sutskever。从自然语言监督中学习可迁移视觉模型。载于*International Conference on Machine Learning*，2021年。 28. Joseph Redmon、Ali Farhadi。YOLO9000：更好、更快、更强。载于*Proceedings of the IEEE conference on computer vision and pattern recognition*，第7263–7271页，2017年。 29. Joseph Redmon、Ali Farhadi。YOLOv3：渐进式改进。*arXiv preprint arXiv:1804.02767*，2018年。 30. 任少卿、何恺明、Ross Girshick、孙剑。Faster R-CNN：基于区域提议网络实现实时目标检测。*Advances in neural information processing systems*，28，2015年。 31. 任少卿、何恺明、Ross Girshick、孙剑。Faster R-CNN：基于区域提议网络实现实时目标检测。*IEEE Transactions on Pattern Analysis and Machine Intelligence*，39(6):1137–1149，2017年。 32. Hamid Rezatofighi、Nathan Tsoi、JunYoung Gwak、Amir Sadeghian、Ian Reid、Silvio Savarese。广义交并比：边界框回归的度量与损失函数。载于*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*，第658–666页，2019年。 33. 邵帅、李子铭、张天元、彭超、余刚、张祥雨、李静、孙剑。Objects365：面向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。

34. Zhiqing Sun, Shengcao Cao, Yiming Yang, and Kris Kitani. Rethinking transformer-based set prediction for object detection. *arXiv preprint arXiv:2011.10881*, 2020.
35. Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9627–9636, 2019.
36. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.
37. Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. *arXiv preprint arXiv:2109.07107*, 2021.
38. Mengde Xu, Zheng Zhang, Han Hu, Jianfeng Wang, Lijuan Wang, Fangyun Wei, Xiang Bai, and Zicheng Liu. End-to-end semi-supervised object detection with soft teacher. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3060–3069, 2021.
39. Zhuyu Yao, Jiangbo Ai, Boxun Li, and Chi Zhang. Efficient detr: Improving end-to-end object detector with dense prior. *arXiv preprint arXiv:2104.01318*, 2021.
40. Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021.
41. Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR 2021: The Ninth International Conference on Learning Representations*, 2021.

34. 孙志清、曹盛操、杨一鸣和Kris Kitani。重新思考基于Transformer的集合预测在目标检测中的应用。*arXiv preprint arXiv:2011.10881*，2020年。 35. 田志、沈春华、陈浩和何通。FCOS：全卷积一阶段目标检测。载于
2019 IEEE/CVF International Conference on Computer Vision (ICCV)，第9627–9636页，2019年。 36. Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Lukasz Kaiser和Illia Polosukhin。注意力机制就是你所需要的一切。载于*Advances in neural information processing systems*，第5998–6008页，2017年。 37. 王英明、张翔宇、杨桐和孙剑。Anchor DETR：基于Transformer检测器的查询设计。*arXiv preprint arXiv:2109.07107*，2021年。 38. 徐梦德、张政、胡涵、王建峰、王丽娟、魏芳云、白翔和刘子成。端到端半监督目标检测与软教师机制。载于
Proceedings of the IEEE/CVF International Conference on Computer Vision，第3060–3069页，2021年。 39. 姚竹雨、艾江波、李伯勋和张驰。高效DETR：利用密集先验改进端到端目标检测器。*arXiv preprint arXiv:2104.01318*，2021年。 40. 袁璐、陈栋栋、陈一灵、Noel Codella、戴熙阳、高剑峰、胡厚东、黄雪东、李博新、李春元等。Florence：计算机视觉的新基础模型。*arXiv preprint arXiv:2111.11432*，2021年。 41. 朱希舟、苏伟杰、陆乐威、李斌、王晓刚和戴继峰。可变形DETR：端到端目标检测中的可变形Transformer。载于*ICLR*
2021: The Ninth International Conference on Learning Representations，2021年。

A Test Time Augmentations (TTA)

We aim to build an end-to-end detector that is free from hand-crafted components. However, to compare with traditional detection models, we also explore the use of TTA in DETR-like models. We only use it in our large model with the SwinL backbone. Our TTA does not obtain an inspiring gain compared with traditional detectors, but we hope our exploration may provide some insights for future studies.

We adopt multi-scale test and horizontal flip as TTA. However, the way of ensembling different augmentations in our method is different from that in traditional methods which usually output duplicate boxes. In traditional methods, the ensembling is done by first gathering predictions from all augmentations and ranked by a confidence score. Then, duplicate boxes are found and eliminated by NMS or box voting. The reason why predictions from all augmentations are gathered first is that duplicate boxes appear not only among different augmentations but also within one augmentation. This ensembling method decreases the performance for our method since DETR-like methods are not prone to output duplicate boxes since their set-based prediction loss inhibits duplicate predictions and ensembling may incorrectly remove true positive predictions [3]. To address this issue, we designed a one-to-one ensembling method. Assume we have n augmentations $Aug_0, Aug_1, ..., Aug_{n-1}$, where Aug_i has predictions $\mathbf{O}^i$ and a pre-defined hyper-parameter weight w^i. $\mathbf{O}^i = \left\{(b_0^i, l_0^i, s_0^i), (b_1^i, l_1^i, s_1^i), ..., (b_{m-1}^i, l_{m-1}^i, s_{m-1}^i)\right\}$ where b_j^i, l_j^i and s_j^i denote the j-th boundbox, label and score, respectively. We let Aug_0 be the main augmentation which is the most reliable one. For each prediction in $\mathbf{O}^0$, we select the prediction with the highest IOU from predictions of each of other augmentations $\mathbf{O}^1, ..., \mathbf{O}^{n-1}$ and make sure the IOU is higher than a predefined threshold. Finally, we ensemble the selected boxes through weighted average as follows

$$b = \frac{1}{\sum I^i} \sum_{i=o}^{n-1} I^i w^i s_{idx(i)}^i b_{idx(i)}^i \tag{3}$$

where $I^i = 1$ when there is at least one box in $\mathbf{O}^i$ with IOU higher than the threshold and $I^i = 0$ otherwise. $idx(i)$ denotes the index of the selected box in $\mathbf{O}^i$.

B Training Efficiency

We provide the GPU memory and training time for our base model in Table 5. All results are reported on 8 Nvidia A100 GPUs with ResNet-50 [14]. The results demonstrate that our models are not only effective but also efficient for training.

A Test Time Augmentations (TTA)

我们的目标是构建一个免于手工设计组件的端到端检测器。然而，为了与传统检测模型进行对比，我们也探索了在类DETR模型中使用测试时增强（TTA）的方法。我们仅在采用SwinL骨干网络的大型模型中应用了这一技术。与传统检测器相比，我们的TTA并未取得显著提升，但我们希望这一探索能为未来研究提供一些启示。

我们采用多尺度测试和水平翻转作为测试时增强（TTA）。然而，本方法中集成不同增强方式的做法与传统方法存在差异——传统方法通常会产生重复检测框。传统方法首先汇集所有增强版本的预测结果，按置信度得分排序，再通过非极大值抑制（NMS）或框投票机制消除重复框。这种先汇集操作的根源在于，重复框不仅会出现在不同增强版本之间，也会存在于单一增强版本内部。由于类DETR方法基于集合预测的损失函数天然抑制重复预测，此类集成方式会降低本方法性能，反而可能误删真实正样本预测[3]。为此，我们设计了一对一集成策略：设有n种增强$Aug_0, Aug_1, ..., Aug_{n-1}$，其中$Aug_i$包含预测结果$\mathbf{O}^i$及预设权重$w^i$。$\mathbf{O}^i = \{(b_0^i, l_0^i, s_0^i), (b_1^i, l_1^i, s_1^i), ..., (b_{m-1}^i, l_{m-1}^i, s_{m-1}^i)\}$中$b_j^i, l_j^i$、$s_j^i$分别表示第$j$个边界框、类别标签和得分。指定最可靠的主增强版本$Aug_0$后，对于$\mathbf{O}^0$中的每个预测，我们从其他各增强版本$\mathbf{O}^1, ..., \mathbf{O}^{n-1}$中选取$IOU$最高的预测，并要求$IOU$超过预设阈值。最终通过加权平均集成所选检测框，公式如下

$$b = \frac{1}{\sum I^i} \sum_{i=o}^{n-1} I^i w^i s_{idx(i)}^i b_{idx(i)}^i \tag{3}$$

当$\mathbf{O}^i$中至少有一个框的IOU高于阈值时，I^i =为1，否则I^i =为0。$idx(i)$表示$\mathbf{O}^i$中所选框的索引。

B Training Efficiency

我们在表5中提供了基础模型的GPU内存占用和训练时间。所有结果均在配备ResNet-50[14]的8块Nvidia A100 GPU上测得。结果表明，我们的模型不仅效果显著，而且训练效率极高。

Model	Batch Size per GPU	Traning Time	GPU Mem.	Epoch	AP
Faster RCNN [30]*	8	~ 60min/ep	13GB	108	42.0
DETR [3]	8	~ 16min/ep	26GB	300	41.2
Deformable DETR [41]*	2	~ 55min/ep	16GB	50	45.4
DINO(Ours)	2	~ 55min/ep	16GB	**12**	**47.9**

Table 5. Training efficieny for different models with ResNet-50 backbone. All models are trianed with 8 Nvidia A100 GPUs. All results are reported by us. * The results of Faster RCNN are tested with the mmdetection framework. * We use the vanilla Deformable DETR without two-stage and bbox refinement during testing.

# Encoder/Decoder	6/6	4/6	3/6	2/6	6/4	6/2	2/4	2/2
AP	47.4	46.2	45.8	45.4	46.0	44.4	44.1	41.2

Table 6. Ablation on the numbers of encoder layers and decoder layers with the ResNet-50 backbone on COCO `val2017`. We use the 12-epoch setting and 100 DN queries without negative samples here.

C Additional Analysis on our Model Components

Analysis on the Number of Encoder and Decoder Layers: We also investigate the influence of varying numbers of encoder and decoder layers. As shown in Table 6, decreasing the number of decoder layers hurts the performance more significantly. For example, using the same 6 encoder layers while decreasing the number of decoder layers from 6 to 2 leads to a 3.0 AP drop. This performance drop is expected as the boxes are dynamically updated and refined through each decoder layer to get the final results. Moreover, we also observe that compared with other DETR-like models like Dynamic DETR [7] whose performance drops by 13.8AP (29.1 vs 42.9) when decreasing the number of decoder layers to 2, the performance drop of DINO is much smaller. This is because our mixed query selection approach feeds the selected boxes from the encoder to enhance the decoder queries. Therefore, the decoder queries are well initialized and not deeply coupled with decoder layer refinement.

# Denoising query	100 CDN	1000 DN	200 DN	100 DN	50 DN	10 DN	No DN
AP	47.9	47.6	47.4	47.4	46.7	46.0	45.1

Table 7. Ablation on number of denoising queries with the ResNet-50 backbone on COCO validation. Note that 100 CND query pairs contains 200 queries which are 100 positive and 100 negative queries.

Analysis on Query Denoising: We continue to investigate the influence of

Model	Batch Size per GPU	Traning Time	GPU Mem.	Epoch	AP
Faster RCNN [30]*	8	$\sim$ 60min/ep	13GB	108	42.0
DETR [3]	8	$\sim$ 16min/ep	26GB	300	41.2
Deformable DETR [41]*	2	$\sim$ 55min/ep	16GB	50	45.4
DINO(Ours)	2	$\sim$ 55min/ep	16GB	**12**	**47.9**

Table 5. 不同模型在ResNet-50骨干网络下的训练效率对比。所有模型均使用8块Nvidia A100 GPU进行训练。结果数据均由我方实测得出。* Faster RCNN的测试结果基于mmdetection框架实现。* 测试阶段我们采用基础版Deformable DETR模型，未启用两阶段检测及边界框优化功能。

# Encoder/Decoder	6/6	4/6	3/6	2/6	6/4	6/2	2/4	2/2
AP	47.4	46.2	45.8	45.4	46.0	44.4	44.1	41.2

Table 6. 在COCO val2017数据集上，基于ResNet-50骨干网络对编码器层数和解码器层数进行消融实验。此处采用12周期设置及100个无负样本的DN查询{v*}。

C Additional Analysis on our Model Components

Analysis on the Number of Encoder and Decoder Layers: 我们还研究了编码器和解码器层数变化的影响。如表6所示，减少解码器层数对性能的损害更为显著。例如，在保持6层编码器的同时，将解码器层数从6层减至2层会导致性能下降3.0个AP点。这种性能下降是预料之中的，因为检测框需通过每一层解码器动态更新与精炼才能获得最终结果。此外，与其他类似DETR的模型（如Dynamic DETR[7]）相比，当解码器层数减至2层时，其性能下降了13.8个AP点（从42.9降至29.1），而DINO的性能下降幅度明显更小。这是因为我们采用的混合查询选择方法将编码器筛选的检测框输入解码器以增强查询特征，使得解码器查询能够获得良好的初始化，而不完全依赖于解码器层的精炼过程。

# Denoising query	100 CDN	1000 DN	200 DN	100 DN	50 DN	10 DN	No DN
AP	47.9	47.6	47.4	47.4	46.7	46.0	45.1

Table 7. 在COCO验证集上使用ResNet-50骨干网络对去噪查询数量进行消融实验。需要注意的是，100对CND查询包含200个查询，其中100个为正查询，100个为负查询。

Analysis on Query Denoising: We continue to investigate the influence of

query denoising by varying the number of denoising queries. We use the optimized dynamic denoising group (detailed in Appendix D.1). As shown in Table 7, when we use less than 100 denoising queries, increasing the number can lead to a significant performance improvement. However, continuing to increase the DN number after 100 yields only a small additional or even worse performance improvement. We also analysis the effect of the number of encoder and decoder Layers in Appendix C.

D More Implementation Details

D.1 Dynamic DN groups

In DN-DETR, all the GT objects (label+box) in one image are collected as one GT group for denoising. To improve the DN training efficiency, multiple noised versions of the GT group in an image are used during training. In DN-DETR, the number of groups is set to five or ten according to different model sizes. As DETR-like models adopt mini-batch training, the total number of DN queries for each image in one batch is padded to the largest one in the batch. Considering that the number of objects in one image in COCO dataset ranges from 1 to 80, this design is inefficient and results in excessive memory consumption. To address this problem, we propose to fix the number of DN queries and dynamically adjust the number of groups for each image according to its number of objects.

D.2 Large-Scale Model Pre-trianing

Objects365 [33] is a large-scale detection data set with over $1.7M$ annotated images for training and $80,000$ annotated images for validation. To use the data more efficiently, We select the first $5,000$ out of $80,000$ validation images as our validation set and add the others to training. We pre-train DINO on Objects365 for 26 epochs using 64 Nvidia A100 GPUs and fine-tune the model on COCO for 18 epochs using 16 Nvidia A100 GPUS. Each GPU has a local batch size of 1 image only. In the fine-tuning stage, we enlarge the image size to $1.5\times$ (i.e., with max size 1200×2000). This adds around 0.5 AP to the final result. To reduce the GPU memory usage, we leverage checkpointing [6] and mixed precision [26] during training. Moreover, we use 1000 DN queries for this large model.

D.3 Other Implementation Details

Basic hyper-parameters. For hyper-parameters, as in DN-DETR, we use a 6-layer Transformer encoder and a 6-layer Transformer decoder and 256 as the hidden feature dimension. We set the initial learning rate (lr) as 1×10^{-4} and adopt a simple lr scheduler, which drops lr at the 11-th, 20-th, and 30-th epoch by multiplying 0.1 for the 12, 24, and 36 epoch settings with RestNet50, respectively. We use the AdamW [16,24] optimizer with weight decay of 1×10^{-4} and train our model on Nvidia A100 GPUs with batch size 16. Since DN-DETR

通过调整去噪查询的数量实现查询去噪。我们采用了优化的动态去噪组（详见附录D.1）。如表7所示，当使用少于100个去噪查询时，增加其数量能带来显著的性能提升。但超过100个后继续增加DN数量，仅能获得微小甚至更差的额外性能改善。我们还分析了编码器与解码器层数的影响，详见附录C。

D More Implementation Details

D.1 Dynamic DN groups

在DN-DETR中，一张图像中的所有真实目标（标签+框）被收集为一个真实组进行去噪。为了提高DN训练效率，训练时会使用图像中真实组的多个噪声版本。DN-DETR根据模型规模的不同，将组数设置为五或十。由于类DETR模型采用小批量训练，每批次中每张图像的DN查询总数会填充至该批次中的最大值。考虑到COCO数据集中单张图像的目标数量在1到80之间变化，这种设计效率低下且导致内存消耗过大。为解决这一问题，我们提出固定DN查询数量，并根据每张图像的目标数量动态调整其组数。

D.2 Large-Scale Model Pre-trianing

Objects365 [33] 是一个大规模检测数据集，包含超过1.7M张标注图像用于训练，以及80,000张标注图像用于验证。为更高效利用数据，我们从80,000张验证图像中选取前5,000张作为验证集，其余则加入训练集。我们使用64块NVIDIA A100 GPU在Objects365上对DINO进行了26轮预训练，随后在COCO数据集上用16块NVIDIA A100 GPU进行了18轮微调。每块GPU的本地批次大小仅为1张图像。在微调阶段，我们将图像尺寸增大至1.5×（（即最大尺寸为1200××2000）），这一操作使最终结果提升了约0.5个AP。为降低GPU内存占用，训练过程中采用了检查点技术[6]和混合精度训练[26]。此外，针对这一大型模型，我们使用了1000个DN查询。

D.3 Other Implementation Details

Basic hyper-parameters. 对于超参数设置，我们与DN-DETR保持一致：采用6层Transformer编码器和6层Transformer解码器，隐藏特征维度设为256。初始学习率(lr)设置为1×10^{-4}，并采用简单的学习率调度策略——在ResNet50框架下分别针对12、24、36轮训练设置，于第11、20、30轮时以0.1倍率逐步降低学习率。优化器选用AdamW[16,24]，权重衰减为1×10^{-4}，训练在NVIDIA A100 GPU上进行，批量大小为16。由于DN-DETR

[17] adopts 300 decoder queries and 3 patterns [37], we use $300 \times 3 = 900$ decoder queries with the same computation cost. Learning schedules of our DINO with SwinL are available in the appendix.

Loss function. We use the L1 loss and GIOU [32] loss for box regression and focal loss [19] with $\alpha = 0.25, \gamma = 2$ for classification. As in DETR [3], we add auxiliary losses after each decoder layer. Similar to Deformable DETR [41], we add extra intermediate losses after the query selection module, with the same components as for each decoder layer. We use the same loss coefficients as in DAB-DETR [21] and DN-DETR [17], that is, 1.0 for classification loss, 5.0 for L1 loss, and 2.0 for GIOU loss.

Detailed model components. We also optimize the detection pipeline used in DAB-DETR [21] and DN-DETR [17]. Following DN-Deformable-DETR [17], we use the same multi-scale approach as in Deformable DETR [41] and adopt the deformable attention. DN-DETR uses different prediction heads with unshared parameters in different decoder layers. In addition, we introduce dynamic denoising group to increase denoising training efficiency and alleviate memory overhead (see Appendix D.1). In this work, we find that using a shared prediction head will add additional performance improvement. This also leads to a reduction of about one million parameters. In addition, we find the conditional queries [25] used in DAB-DETR does not suit our model and we do not include them in our final model.

Training augmentation. We use the same random crop and scale augmentation during training following DETR [3]. For example, we randomly resize an input image with its shorter side between 480 and 800 pixels and its longer side at most 1333. For DINO with SwinL, we pre-train the model using the default setting, but finetune using 1.5× larger scale (shorter side between 720 and 1200 pixels and longer side at most 2000 pixels) to compare with models on the leaderboard [1]. Without using any other tricks, we achieve the result of 63.1 on `val2017` and 63.2 on `test-dev` without test time augmentation (TTA) (see Appendix A), outperforming the previous state-of-the-art result 63.1 achieved by SwinV2 [22] with a much neater solution.

Multi-scale setting. For our 4-scale models, we extract features from stages 2, 3, and 4 of the backbone and add an extra feature by down-sampling the output of the stage 4. An additional feature map of the backbone stage 1 is used for our 5-scale models. For hyper-parameters, we set $\lambda_1 = 1.0$ and $\lambda_2 = 2.0$ and use 100 CDN pairs which contain 100 positive queries and 100 negative queries.

D.4 Detailed Hyper-parameters

We list the hyper-parameters for those who want to reproduce our results in Table 8.

[17]采用了300个解码器查询和3种模式[37]，而我们使用300×3=900解码器查询，保持相同的计算成本。我们基于SwinL的DINO学习计划详见附录。

Loss function. 我们采用L1损失和GIOU[32]损失进行边界框回归，并使用焦点损失[19]进行分类，其参数设置为α =0.25, γ =2。如DETR[3]所述，我们在每个解码器层后添加了辅助损失。与可变形DETR[41]类似，我们在查询选择模块后额外增加了中间损失，其构成与各解码器层的损失相同。损失系数遵循DAB-DETR[21]和DN-DETR[17]的设置，即分类损失系数为1.0，L1损失系数为5.0，GIOU损失系数为2.0。

Detailed model components. 我们还优化了DAB-DETR[21]和DN-DETR[17]中使用的检测流程。遵循DN-Deformable-DETR[17]的做法，我们采用与Deformable DETR[41]相同的多尺度方法，并运用了可变形注意力机制。DN-DETR在不同解码器层使用参数不共享的独立预测头。此外，我们引入了动态去噪组以提升去噪训练效率并降低内存开销（详见附录D.1）。本研究发现，采用共享预测头能带来额外的性能提升，同时减少约百万参数。我们还发现DAB-DETR采用的条件查询[25]并不适配当前模型，故未将其纳入最终模型。

Training augmentation. 我们遵循DETR[3]的方法，在训练期间采用相同的随机裁剪和尺度增强策略。例如，我们会随机调整输入图像的尺寸，使其短边介于480至800像素之间，长边不超过1333像素。对于采用SwinL的DINO模型，预训练阶段使用默认设置，但在微调时采用1.5×倍的更大尺度（短边720至1200像素，长边不超过2000像素），以便与排行榜[1]上的模型进行对比。在不使用任何其他技巧的情况下，我们在val2017上取得了63.1的成绩，在test-dev上达到63.2（未使用测试时增强TTA，详见附录A），以更为简洁的方案超越了此前由SwinV2[22]实现的63.1的先进水平。

Multi-scale setting. 对于我们的4尺度模型，我们从主干网络的第2、3、4阶段提取特征，并通过下采样第4阶段的输出添加一个额外特征。而在5尺度模型中，我们还利用了主干网络第1阶段的额外特征图。在超参数设置上，我们设定λ_1 =1.0和λ_2 =2.0，并采用包含100个正查询和100个负查询的100对CDN组合。

D.4 Detailed Hyper-parameters

我们在表8中列出了希望复现我们结果所需的超参数。

Item	Value
lr	0.0001
lr_backbone	1e-05
weight_decay	0.0001
clip_max_norm	0.1
pe_temperature	20
enc_layers	6
dec_layers	6
dim_feedforward	2048
hidden_dim	256
dropout	0.0
nheads	8
num_queries	900
enc_n_points	4
dec_n_points	4
transformer_activation	“relu”
batch_norm_type	“FrozenBatchNorm2d”
set_cost_class	2.0
set_cost_bbox	5.0
set_cost_giou	2.0
cls_loss_coef	1.0
bbox_loss_coef	5.0
giou_loss_coef	2.0
focal_alpha	0.25
dn_box_noise_scale	0.4
dn_label_noise_ratio	0.5

Table 8. Hyper-parameters used in our models.

Item	Value
lr	0.0001
lr_backbone	1e-05
weight_decay	0.0001
clip_max_norm	0.1
pe_temperature	20
enc_layers	6
dec_layers	6
dim_feedforward	2048
hidden_dim	256
dropout	0.0
nheads	8
num_queries	900
enc_n_points	4
dec_n_points	4
transformer_activation	“relu”
batch_norm_type	“FrozenBatchNorm2d”
set_cost_class	2.0
set_cost_bbox	5.0
set_cost_giou	2.0
cls_loss_coef	1.0
bbox_loss_coef	5.0
giou_loss_coef	2.0
focal_alpha	0.25
dn_box_noise_scale	0.4
dn_label_noise_ratio	0.5

Table 8. 我们模型中使用的超参数。