DN-DETR: Accelerate DETR Training by Introducing Query DeNoising

Feng Li*, Hao Zhang*, Shilong Liu, Jian Guo, Lionel M. Ni, and Lei Zhang, *IEEE Fellow*

Abstract—We present in this paper a novel denoising training method to speed up DETR (DEtection TRansformer) training and offer a deepened understanding of the slow convergence issue of DETR-like methods. We show that the slow convergence results from the instability of bipartite graph matching which causes inconsistent optimization goals in early training stages. To address this issue, except for the Hungarian loss, our method additionally feeds GT bounding boxes with noises into the Transformer decoder and trains the model to reconstruct the original boxes, which effectively reduces the bipartite graph matching difficulty and leads to faster convergence. Our method is universal and can be easily plugged into any DETR-like method by adding dozens of lines of code to achieve a remarkable improvement. As a result, our DN-DETR results in a remarkable improvement ($+1.9$AP) under the same setting and achieves 46.0 AP and 49.5 AP trained for 12 and 50 epochs with the ResNet-50 backbone. Compared with the baseline under the same setting, DN-DETR achieves comparable performance with 50% training epochs. We also demonstrate the effectiveness of denoising training in CNN-based detectors (Faster R-CNN), segmentation models (Mask2Former, Mask DINO), and more DETR-based models (DETR, Anchor DETR, Deformable DETR). Code is available at https://github.com/IDEA-Research/DN-DETR.

Index Terms—Object Detection, Vision Transformer, DETR, Model Convergence, Denoising Training

1 Introduction

Object detection is a fundamental task in computer vision that aims to predict the bounding boxes and classes of objects in an image. While having made remarkable progress, classical detectors [18], [17] were mainly based on convolutional neural networks, until Carion *et al.* [1] recently introduced Transformers [20] into object detection and proposed DETR (DEtection TRansformer).

In contrast to previous detectors, DETR uses learnable queries to probe image features from the output of Transformer encoders and bipartite graph matching to perform set-based box prediction. Such a design effectively eliminates hand-designed anchors and non-maximum suppression (NMS) and makes object detection end-to-end optimizable. However, DETR suffers from prohibitively slow training convergence compared with previous detectors. To obtain a good performance, it usually takes 500 epochs of training on the COCO detection dataset, in contrast to 12 epochs used in the original Faster-RCNN training.

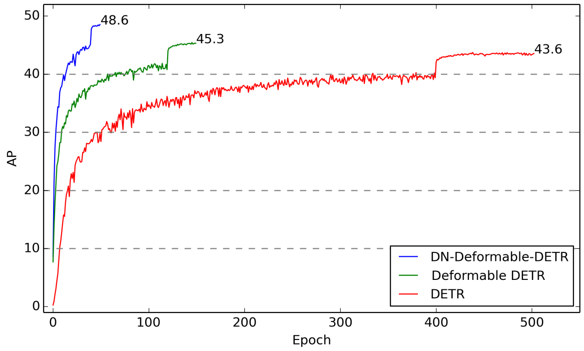


Fig. 1. Convergence curve between our model DN-Deformable-DETR built upon Deformable DETR with denoising training and previous models under ResNet-50 backbone.

Much work [21], [15], [25], [19], [14], [6] has tried to identify the root cause and mitigate the slow convergence issue. Some of them address the problem by improving the model architecture. For example, Sun *et al.* [19] attributed the slow convergence issue to the low efficiency of the cross-attention and proposed an encoder-only DETR. Dai *et al.* [6] designed an RoI-based dynamic decoder to help the decoder focus on regions of interest. More recent works propose to associate each DETR query with a specific spatial position rather than multiple positions for more efficient feature probing [21], [15], [25], [14]. For instance, Conditional DETR [15] decouples each query into a content part and a positional part, enforcing a query to have a clear correspondence with a specific spatial position. Deformable DETR [25] and Anchor DETR [21] directly treat $2D$ reference points as queries to perform cross-attention. DAB-DETR [14] interprets queries as 4-D anchor boxes and learns to progressively improve them layer by layer.

Despite all the progress, few works pay attention to the bipartite graph matching part for more efficient training. In this study, we find that the slow convergence issue also results from the discrete bipartite graph matching component, which is unstable especially in the early stages of training due to the nature of stochastic optimization. As a consequence, for the same image, a query is often

- *Feng Li and Hao Zhang are with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong.*
- *Shilong Liu is with the Department of Computer Science and Engineering, Tsinghua University, Beijing.*
- *Lionel Ni is the president of The Hong Kong University of Science and Technology (Guangzhou).*
- *Jian Guo and Lei Zhang are with IDEA.*
- ** denotes equal contribution.*

DN-DETR：通过引入查询去噪加速DETR训练

李峰*、张浩*、刘世龙、郭健、倪明选*IEEE Fellow*和张磊

摘要—本文提出了一种新颖的去噪训练方法，旨在加速DETR（DEtection TRansformer）的训练过程，并深化对DETR类方法收敛速度缓慢问题的理解。我们揭示了收敛缓慢源于二分图匹配的不稳定性，这种不稳定性导致早期训练阶段优化目标不一致。为解决该问题，除匈牙利损失外，我们的方法还向Transformer解码器注入带噪声的GT边界框，并训练模型重建原始框，从而有效降低二分图匹配难度并加速收敛。该方法具有普适性，仅需添加数十行代码即可轻松集成到任何DETR类方法中，实现显著提升。实验表明，在相同设置下，DN-DETR取得显著改进（+1.9AP），使用ResNet-50骨干网络训练12和50周期分别达到46.0 AP和49.5 AP。与同设置基线相比，DN-DETR仅需50%训练周期即可达到相当性能。我们还验证了去噪训练在CNN检测器（Faster R-CNN）、分割模型（Mask2Former、Mask DINO）及其他DETR变体（DETR、Anchor DETR、Deformable DETR）中的有效性。代码已开源：https://github.com/IDEA-Research/DN-DETR。

索引术语—目标检测，视觉变换器，DETR，模型收敛，去噪训练

1 引言

目标检测是计算机视觉中的一项基础任务，旨在预测图像中物体的边界框和类别。尽管传统检测器[18][17]基于卷积神经网络取得了显著进展，但直到Carion {v*} [1]近期将Transformer[20]引入目标检测领域并提出DETR（DEtection TRansformer）模型，才实现了方法论的革新。

与之前的检测器不同，DETR采用可学习的查询向量从Transformer编码器输出中探查图像特征，并利用二分图匹配进行基于集合的边界框预测。这一设计有效消除了手工设计的锚框和非极大值抑制（NMS）步骤，使目标检测任务能够端到端优化。然而，DETR的训练收敛速度相比传统检测器显著缓慢——在COCO检测数据集上通常需要500轮训练才能达到良好性能，而原始Faster-RCNN仅需12轮训练。

大量研究[21]、[15]、[25]、[19]、[14]、[6]致力于探究收敛速度缓慢的根本原因并寻求缓解方案。部分工作通过改进模型架构来解决该问题。例如，Sun*et al.*[19]将收敛缓慢归因于交叉注意力效率低下，并提出了一种仅含编码器的DETR模型。Dai*et al.*[6]则设计了基于感兴趣区域(RoI)的动态解码器，使解码器能聚焦于关键区域。最新研究则尝试通过关联...

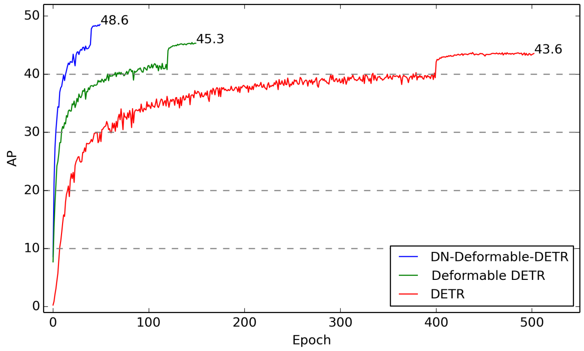


图1. 基于Deformable DETR并采用去噪训练的我们的模型DN-Deformable-DETR与先前模型在ResNet-50骨干网络下的收敛曲线对比。

每个DETR查询对应一个特定的空间位置，而非多个位置，以实现更高效的特征探测[21], [15], [25], [14]。例如，条件DETR[15]将每个查询解耦为内容部分和位置部分，强制查询与特定空间位置建立明确对应关系。可变形DETR[25]和锚点DETR[21]直接将2{v*}个参考点作为查询执行交叉注意力。DAB-DETR[14]将查询解释为4维锚框，并学习逐层渐进式优化它们。

尽管取得了诸多进展，但很少有研究关注二分图匹配部分以实现更高效的训练。在本研究中，我们发现收敛速度慢的问题同样源于离散二分图匹配组件——由于随机优化的本质特性，该组件在训练初期尤其不稳定。其结果是，对于同一幅图像，查询往往会

- *Feng Li and Hao Zhang are with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong.*
- *Shilong Liu is with the Department of Computer Science and Engineering, Tsinghua University, Beijing.*
- *Lionel Ni is the president of The Hong Kong University of Science and Technology (Guangzhou).*
- *Jian Guo and Lei Zhang are with IDEA.*
- ** denotes equal contribution.*

matched with different objects in different epochs, which makes optimization ambiguous and inconstant.

To address this problem, we propose a novel training method by introducing a query denoising task to help stabilize bipartite graph matching in the training process. Since previous works have shown effectiveness in interpreting queries as reference points [25], [21] or anchor boxes [14], which contain positional information, we follow their viewpoint and use 4D anchor boxes as queries. Our solution is to feed noised GT bounding boxes as noised queries together with learnable anchor queries into Transformer decoders. Both kinds of queries have the same input format of (x, y, w, h) and can be fed into Transformer decoders simultaneously. For noised queries, we perform a denoising task to reconstruct their corresponding GT boxes. For other learnable anchor queries, we use the same training loss and bipartite matching as in the vanilla DETR. As the noised bounding boxes do not need to go through the bipartite graph matching component, the denoising task can be regarded as an easier auxiliary task, helping DETR alleviate the unstable discrete bipartite matching and learn bounding box prediction more quickly. Meanwhile, the denoising task also helps lower the optimization difficulty because the added random noise is usually small. To maximize the potential of this auxiliary task, we also regard each decoder query as a bounding box + a class label embedding so that we are able to conduct both box denoising and label denoising.

In summary, our method is a denoising training approach. Our loss function consists of two components. One is a reconstruction loss and the other is a Hungarian loss which is the same as in other DETR-like methods. Our method can be easily plugged into any existing DETR-like method. For convenience, we utilize DAB-DETR [14] to evaluate our method since their decoder queries are explicitly formulated as 4D anchor boxes (x, y, w, h). For DETR variants that only support 2D anchor points such as anchor DETR [21], we can do denoising on anchor points. For those that do not support anchors like the vanilla DETR [1], we can do linear transformation to map 4D anchor boxes to the same latent space as for other learnable queries.

To the best of our knowledge, this is the first work to introduce the denoising principle into detection models. We summarize our contribution as follows:

1) We design a novel training method to speed up DETR training. Experimental results show that our method not only accelerates training convergence but also leads to a remarkably better training result — achieving the best result among all detection algorithms in the 12-epoch setting. Moreover, our method shows a remarkable improvement (+**1.9** AP) over our baseline DAB-DETR and can be easily integrated into other DETR-like methods.
2) We analyze the slow convergence of DETR from a novel viewpoint and give a deeper understanding of DETR training. We design a metric to evaluate the instability of bipartite matching and verify that our method can effectively lower the instability.
3) We conduct a series of ablation studies to analyze the effectiveness of different components of our model, such as noise, label embedding, and attention mask.

This paper is an extension of our previous paper [10] that was accepted to CVPR'2022 as an oral presentation. Compared with its conference version, this paper brings some new contributions as follows.

1) We achieve better results and faster convergence by introducing deformable attention into our decoder layer.
2) We further demonstrate the effectiveness of denoising training by adding it to other DETR-like models without 4D anchor design, including Vanilla DETR without explicit anchors and Anchor DETR with only 2D anchors. We also show denoising training can improve segmentation models such as Mask2Former and Mask DINO.
3) We incorporate denoising training to the traditional CNN detector Faster R-CNN to show its generalization ability.
4) We provide more experimental results and analysis to get a better understanding of our method.

2 Related Work

2.1 Classical CNN Detectors

Most modern object detection models are based on convolutional networks, which have achieved significant success in recent years. Classical CNN-based detectors can be divided into 2 categories, one-stage, and two-stage methods. Two-stage methods like HTC [2] and Fast R-CNN [8] first generate some region proposals and then decide whether each region contains an object and do bounding box regression to get a refined box. Ren *et al.* [18] proposed an end-to-end method that utilizes a Region Proposal Network to predict anchor boxes. In contrast to two-stage methods, one-stage methods, including YOLO900 [16] and YOLOv3 [17] directly predict the offset of real boxes relative to anchor boxes.

Though these methods achieve top performance on many datasets, they are sensitive to the way how anchors are generated. In addition, they require some hand-crafted components like non-maximum suppression (NMS) and label assignment rules. Therefore, they suffer from these drawbacks and can not be end-to-end optimized.

2.2 DETR-based Detectors

Carion *et al.* [1] proposed an end-to-end object detector based on Transformers [20] named DETR (DEtection TRansformer) without using anchors. While DETR achieves comparable results with Faster-RCNN [18], its training suffers severely from the slow convergence problem — it needs 500 epochs of training to obtain a good performance.

Many recent works have attempted to speed up the training process of DETR. Some find the cross attention of Transformer decoders in DETR inefficient and make improvements in different ways. For example, Dai *et al.* [**?**] designed a dynamic decoder that can focus on regions of interest in a coarse-to-fine manner and lower the learning difficulty. Sun *et al.* [19] discarded the Transformer decoder and proposed an encoder-only DETR. Another series

在不同时期与不同对象匹配，这使得优化过程模糊且不稳定。

为解决这一问题，我们提出了一种新颖的训练方法，通过引入查询去噪任务来帮助稳定训练过程中的二分图匹配。由于先前研究已证明将查询解释为包含位置信息的参考点[25][21]或锚框[14]具有有效性，我们遵循其观点，采用4D锚框作为查询。我们的解决方案是将加噪的真实边界框作为噪声查询与可学习的锚查询一同输入Transformer解码器。两类查询均采用相同的输入格式(x, y, w, h)，可并行馈入解码器。对于噪声查询，我们执行去噪任务以重建其对应的真实框；对于其他可学习锚查询，则沿用原始DETR的匹配损失与二分图匹配机制。由于加噪边界框无需经过二分图匹配环节，该去噪任务可视为更简单的辅助任务，既能帮助DETR缓解不稳定的离散匹配问题，又能加速边界框预测的学习。同时，由于添加的随机噪声通常较小，去噪任务也有助于降低优化难度。为充分发挥该辅助任务的潜力，我们还将每个解码器查询视为边界框+与类别标签嵌入的组合，从而能够同步执行框去噪与标签去噪。

总之，我们的方法是一种去噪训练策略。损失函数由两部分组成：一是重构损失，二是与其他类DETR方法相同的匈牙利损失。该方法可轻松嵌入任何现有类DETR框架。为方便起见，我们采用DAB-DETR[14]评估本方法，因其解码器查询被显式表述为4D锚框(x, y, w, h)。对于仅支持2D锚点的类DETR变体（如anchor DETR[21]），我们可在锚点上实施去噪；而对于不支持锚点的原始DETR[1]，则可通过线性变换将4D锚框映射至与其他可学习查询相同的潜在空间。

据我们所知，这是首次将去噪原理引入检测模型的研究工作。我们将贡献总结如下：

1) 我们设计了一种新颖的训练方法来加速DETR的训练。实验结果表明，该方法不仅加快了训练收敛速度，还显著提升了训练效果——在12轮训练设置下取得了所有检测算法中的最佳成绩。此外，我们的方法相较于基线DAB-DETR展现出显著提升（+1.9 AP），并能轻松集成到其他类DETR方法中。

2) 我们从全新视角分析了DETR收敛缓慢的问题，并深化了对DETR训练的理解。我们设计了一项指标来评估二分匹配的不稳定性，并验证了我们的方法能有效降低这种不稳定性。

3) 我们进行了一系列消融实验，以分析我们{v*}不同组件的有效性模型，如噪声、标签嵌入和注意力掩码{v*}。

本文是我们先前被CVPR’2022作为口头报告接受的论文[10]的扩展版本。与会议版本相比，本文带来了以下新贡献。

1) 通过在解码器层引入可变形注意力机制，我们实现了更优的结果与更快的收敛速度。 2) 我们进一步验证了去噪训练的有效性：将其应用于其他无4D锚点设计的类DETR模型（包括无显式锚点的Vanilla DETR和仅含2D锚点的Anchor DETR）时，性能显著提升。同时证明该训练方式可改善Mask2Former、Mask DINO等分割模型的效果。 3) 我们将去噪训练整合至传统CNN检测器Faster R-CNN，以展示其泛化能力。 4) 通过补充大量实验结果与分析，我们更深入地阐释了本方法的优势。 （注：公式标记{v*}按原文保留未翻译）

2 相关工作

2.1 经典CNN检测器

大多数现代目标检测模型基于卷积网络，近年来取得了显著成功。经典的基于CNN的检测器可分为两类：单阶段与双阶段方法。双阶段方法如HTC[2]和Fast R-CNN[8]首先生成若干区域提议，随后判断每个区域是否包含目标并进行边界框回归以获得精细化框。Ren *et al.*[18]提出了一种端到端方法，利用区域提议网络预测锚框。与双阶段方法不同，单阶段方法（包括YOLO900[16]和YOLOv3[17]）直接预测真实框相对于锚框的偏移量。

尽管这些方法在许多数据集上达到了顶尖性能，但它们对锚点生成的方式非常敏感。此外，它们还需要一些人工设计的组件，如非极大值抑制（NMS）和标签分配规则。因此，这些方法受限于上述缺点，无法实现端到端的优化。

2.2 基于DETR的检测器

Carion *et al.* [1] 提出了一种基于Transformers [20]的端到端目标检测器，名为DETR（DEtection TRansformer），无需使用锚框。尽管DETR取得了与Faster-RCNN [18]相当的结果，但其训练过程深受收敛速度缓慢的问题困扰——需要500个训练周期才能获得良好性能。

近期许多研究尝试加速DETR的训练过程。部分学者发现DETR中Transformer解码器的交叉注意力机制效率低下，并从不同角度进行改进。例如Dai *et al.*[?]设计了动态解码器，能以由粗到精的方式聚焦感兴趣区域，降低学习难度；Sun *et al.*[19]则摒弃Transformer解码器，提出纯编码器架构的DETR。另一类研究

of works make improvements in decoder queries. Zhu *et al.* [25] designed an attention module that only attends to some sampling points around a reference point. Meng et al. [15] decoupled each decoder query into a content part and a position part and only utilized the content-to-content and position-to-position terms in the cross-attention formulation. Yao *et al.* [22] utilized a Region Proposal Network (RPN) to propose top-K anchor points. DAB-DETR [14] uses 4-D box coordinates as queries and updates boxes layer by layer in a cascade manner.

Despite all the progress, none of them treats bipartite graph matching used in the Hungarian loss as the main reason for slow convergence. Sun *et al.* [19] analyzed the impact of Hungarian loss by using a pre-trained DETR as a teacher to provide the GT label assignment for a student model and train the student model. They found that the label assignment only helps the convergence in the early stage of training but does not influence the final performance significantly. Therefore, they concluded that the Hungarian loss is not the main reason for the slow convergence. In this work, we give a different analysis with an effective solution that leads to a different conclusion.

We adopt DAB-DETR as the basic detection architecture to evaluate our training method, where the label embedding appended with an indicator is used to replace the decoder embedding part to support label denoising. The difference between our method and other methods is mainly in the training method. In addition to the Hungarian loss, we add a denoising loss as an easier auxiliary task that can accelerate training and boost performance significantly. Chen *et al.* [4] augments their sequence with synthetic noise objects, but is totally different from our method. They set the targets of noise objects to the ”noise” class (not belonging to any ground-truth classes) so that they can delay the End-of-Sentence (EOS) token and improve the recall. In contrast to their method, we set the target of noised boxes to the original boxes, and the motivation is to bypass bipartite graph matching and directly learn to approximate ground-truth boxes.

We are pleased to see that many very recent detection models adopt our proposed denoising training to accelerate convergence for detection and segmentation models, such as DINO [24], Mask DINO [11], Group DETR [3], and SAM-DETR++ [23]. DINO [24] further develops our denoising training by feeding hard-negative samples and training the model to reject them. Therefore, the proposed Contrastive Denoising (CDN) further improves the performance. Mask DINO [11] extends denoising to three image segmentation tasks (instance, panoptic, and semantic) by reconstructing masks from noised boxes. Group DETR [3] and SAM-DETR+++[23] also adopt denoising training in their model to achieve better performance. These models demonstrate the effectiveness and generalization capabilities of our methods.

3 Why Denoising accelerates DETR training?

3.1 Stablize Hungarian Mathcing

Hungarian matching is a popular algorithm in graph matching. Given a cost matrix, the algorithm outputs an optimal matching result. DETR is the first algorithm that adopts Hungarian matching in object detection to solve the matching problem between predicted objects and ground-truth objects. DETR turns ground-truth assignment into a dynamic process, which brings in an instability problem due to its discrete bipartite matching and the stochastic training process. There are works [7] showing that Hungarian matching does not result in stable matching since blocking pairs exist. A small change in the cost matrix may cause an enormous change in the matching result, which will further lead to inconsistent optimization goals for decoder queries.

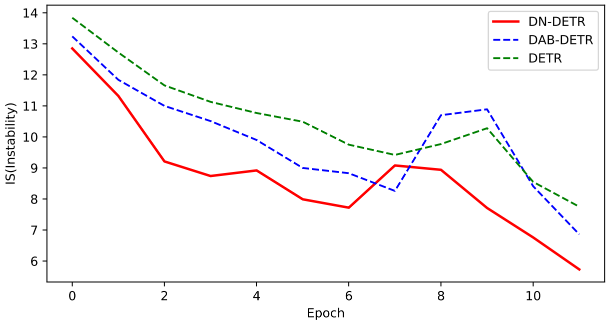


Fig. 2. The IS of DAB-DETR and DN-DETR during training. For each method, we train 12 epoch on the same setting. We test the change of the Hungarian matching between each two epochs on the Validation set as the IS.

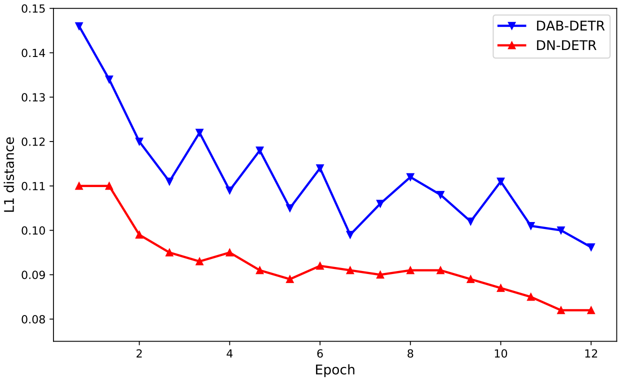


Fig. 3. A comparison of DAB-DETR and DN-DETR on anchor-target distance.

We view the training process of DETR-like models as two stages, learning “good anchors” and learning relative offsets. Decoder queries are responsible for learning anchors as shown in previous works [14] and [25]. The inconsistent update of anchors can make it difficult to learn relative offsets. Therefore, in our method, we leverage a denoising task as a training shortcut to make relative offset learning easier, as the denoising task bypasses bipartite matching. Since we interpret each decoder query as a 4-D anchor box, a noised query can be regarded as a “good anchor” which has a corresponding ground-truth box nearby. The denoising training thus has a clear optimization goal - to predict the original bounding box, which essentially avoids the ambiguity brought by Hungarian matching.

To quantitatively evaluate the instability of the bipar-

在解码器查询方面，多项工作提出了改进。Zhu *et al.* [25]设计了一种注意力模块，仅关注参考点周围的若干采样点。Meng等人[15]将每个解码器查询解耦为内容部分和位置部分，并在交叉注意力公式中仅利用内容对内容及位置对位置的项。Yao *et al.* [22]采用区域提议网络（RPN）生成top-K锚点。DAB-DETR[14]则以4维框坐标作为查询，并以级联方式逐层更新检测框。

尽管取得了诸多进展，但现有研究均未将匈牙利损失中使用的二分图匹配视为收敛缓慢的主要原因。Sun *et al.* [19]通过使用预训练DETR作为教师模型为学生模型提供GT标签分配并训练学生模型，分析了匈牙利损失的影响。他们发现标签分配仅在训练初期有助于收敛，但对最终性能影响甚微。因此，他们得出结论认为匈牙利损失并非收敛缓慢的主因。本研究通过提出有效解决方案给出了不同分析，并得出了相异结论。

我们采用DAB-DETR作为基础检测架构来评估训练方法，其中通过附加指示符的标签嵌入替代解码器嵌入部分，以支持标签去噪。与其他方法的主要差异在于训练策略：除了匈牙利损失外，我们新增了去噪损失作为更简单的辅助任务，可显著加速训练并提升性能。Chen *et al.* [4]通过合成噪声对象增强序列，但与本方法存在本质区别——他们将噪声对象目标设为"噪声"类（不属于任何真实类别）以延迟句子结束符(EOS)并提高召回率；而本方法则将噪声框目标设为原始框，其核心动机是绕过二分图匹配，直接学习逼近真实框。

我们欣喜地发现，许多最新检测模型采用了我们提出的去噪训练方法，以加速检测和分割模型的收敛速度，例如DINO [24]、Mask DINO[11]、Group DETR[3]和SAM-DETR++[23]。DINO[24]通过输入困难负样本并训练模型拒绝这些样本，进一步发展了我们的去噪训练方法。因此，所提出的对比去噪（CDN）进一步提升了性能。Mask DINO[11]通过从带噪框重建掩码，将去噪方法扩展到三种图像分割任务（实例、全景和语义）。Group DETR[3]和SAM-DETR+++[23]也在其模型中采用去噪训练以获得更好的性能。这些模型证明了我们方法的有效性和泛化能力。

3 为什么去噪能加速DETR训练?

3.1 稳定匈牙利匹配

匈牙利匹配是图匹配中一种流行算法。给定一个成本矩阵，该算法输出一个最优解{v*}。

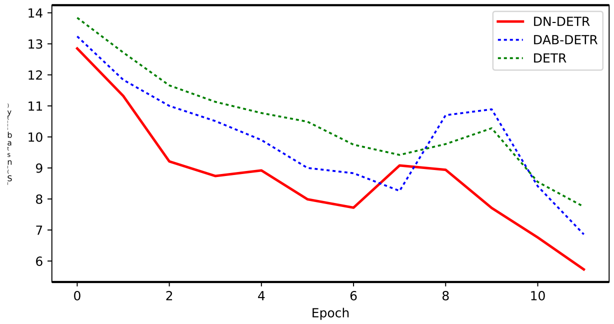


图2. DAB-DETR与DN-DETR训练过程中的IS。每种方法均在同一设置下训练12个周期。我们以验证集上每两周期间匈牙利匹配的变化作为IS进行测试。

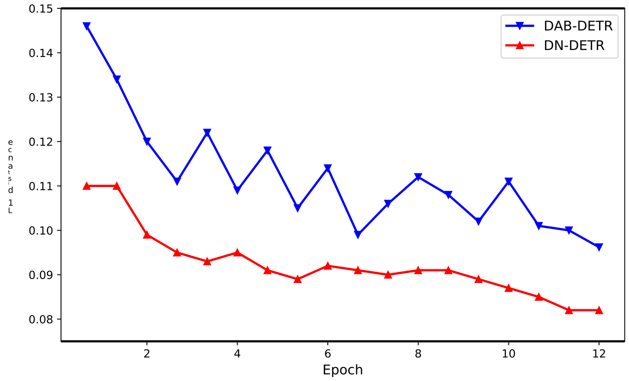


图3. DAB-DETR与DN-DETR在锚点-目标距离上的比较。

最优匹配结果。DETR是首个在目标检测中采用匈牙利匹配算法来解决预测对象与真实标注对象间匹配问题的算法。DETR将真实标注分配转变为动态过程，由于其离散二分匹配和随机训练过程的特性，引入了不稳定性问题。已有研究[7]表明，由于阻塞对的存在，匈牙利匹配无法实现稳定匹配。成本矩阵的微小变化可能导致匹配结果发生巨大改变，进而导致解码器查询的优化目标不一致。

我们将类似DETR模型的训练过程视为两个阶段：学习“优质锚点”与学习相对偏移量。如先前研究[14]和[25]所示，解码器查询负责学习锚点。锚点的不一致更新会增加相对偏移量学习的难度。因此，在本方法中，我们利用去噪任务作为训练捷径来简化相对偏移量的学习——因为去噪任务绕过了二分图匹配。由于我们将每个解码器查询解释为一个4维锚框，带噪声的查询可视为附近存在对应真实框的“优质锚点”。这种去噪训练因而具有明确的优化目标：预测原始边界框，从根本上避免了匈牙利匹配带来的歧义性。

为了定量评估双足机器人的不稳定性

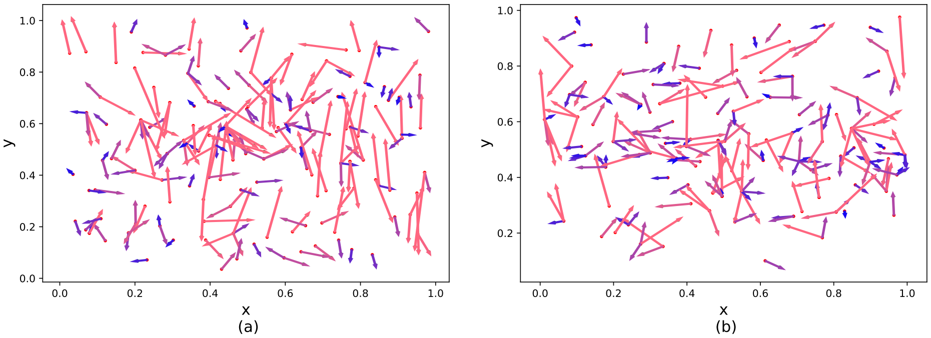


Fig. 4. (a)(b)Some examples of anchors and targets for DAB-DETR and DN-DETR, respectively. Each arrow starts from an anchor and points to a target. The color of each arrow shows its l_1 length and cooler colors denote shorter arrows.

tite matching result, we design a metric as follows. For a training image, we denote the predicted objects from Transformer decoders as $\mathbf{O^i} = \{O_0^i, O_1^i, ..., O_{N-1}^i\}$ in the i-th epoch, where N is the number of predicted objects, and the ground-truth objects as $\mathbf{T} = \{T_0, T_1, T_2, ..., T_{M-1}\}$ where M is the number of ground-truth objects. After bipartite matching, we compute an index vector $\mathbf{V^i} = \{V_0^i, V_1^i, ..., V_{N-1}^i\}$ to store the matching result of epoch i as follows.

$$V_n^i = \begin{cases} m, & \text{if } O_n^i \text{ matches } T_m \\ -1, & \text{if } O_n^i \text{ matches nothing} \end{cases} \tag{1}$$

We define the instability of epoch i for one training image as the difference between its V^i and V^{i-1}, which is calculated as

$$IS^i = \sum_{j=0}^{N} \mathbb{1}(V_n^i \neq V_n^{i-1}) \tag{2}$$

where $\mathbb{1}(\cdot)$ is the indicator function. $\mathbb{1}(x) = 1$ if x is true and 0 otherwise. The instability of epoch i for the whole data set is averaged over the instability numbers for all images. We omit the index for an image for notation simplicity in Eq. (1) and Eq. (2).

Fig. 3 shows a comparison of IS between our DN-DETR (DeNoising DETR) and DAB-DETR. We conduct this evaluation on the COCO 2017 validation set [13], which has 7.36 objects per image on average. So the largest possible IS is $7.36 \times 2 = 14.72$. Fig. 3 clearly shows that our method effectively alleviates the instability of matching.

3.2 Make Query Search More Locally

We also show that DN-DETR can help detection by reducing the distance between anchors and the corresponding targets. DETR [1] shows from the visualization that its positional queries have several operating modes, which makes a query search from a wide region for a predicted box. However, DN-DETR has much smaller mean distances between initial anchors (positional queries) and targets. As shown in Fig. 4(a), we compute the mean l_1 distance between initial anchors and the matched ground-truth boxes in the last decoder layer for DAB-DETR and our model.

As denoising training trains the model to reconstruct boxes from the noised ones that are close to the ground truth, the model will search more locally for prediction, which makes each query focus on regions nearby and prevents potential prediction conflicts between queries. Fig. 4(b) and (c) are some examples of anchors and targets in DAB-DETR and DN-DETR. Each arrow starts from an anchor and ends with its matched ground-truth box. We use color to reflect the length of the arrows. The shortened distances between anchors and targets make the training process easier and therefore converge faster.

4 DN-DETR

4.1 Overview

We base on the architecture of DAB-DETR [14] to implement our training method. Similar to DAB-DETR, we explicitly formulate the decoder queries as box coordinates. The only difference between our architecture and theirs lies in the decoder embedding, which is specified as class label embedding to support label denoising. Our main contribution is the training method as shown in Fig. 6.

Similar to DETR, our architecture contains a Transformer encoder and a Transformer decoder. On the encoder side, the image features are extracted with a CNN backbone and then fed into the Transformer encoder with positional encodings to attain refined image features. On the decoder side, queries are fed into the decoder to search for objects through cross-attention.

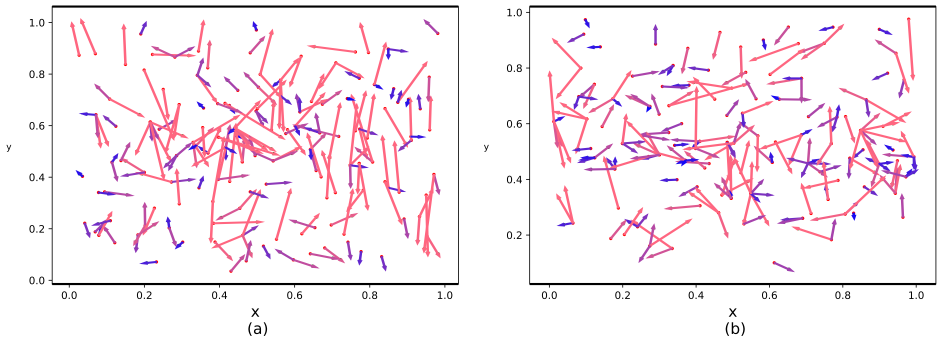
We denote decoder queries as $\mathbf{q} = \{q_0, q_1, ..., q_{N-1}\}$ and the output of the Transformer decoder as $\mathbf{o} = \{o_0, o_1, ..., o_{N-1}\}$. We also use F and A to denote the refined image features after the Transformer encoder, and the attention mask derived based on the denoising task design. We can formulate our method as follows.

$$\mathbf{o} = D(\mathbf{q}, F|A) \tag{3}$$

where D denotes the Transformer decoder.

There are two parts to decoder queries. One is the matching part. The inputs of this part are learnable anchors, which are treated in the same way as in DETR. That is, the matching part adopts bipartite graph matching and learns to approximate the ground-truth box-label pairs with matched decoder outputs. The other is the denoising part. The inputs of this part are noised ground-truth (GT) box-label pairs which are called **GT objects** in the rest of the paper. The outputs of the denoising part aim to reconstruct GT objects.

In the following, we abuse the notations to denote the denoising part as $\mathbf{q} = \{q_0, q_1, ..., q_{K-1}\}$ and the matching



(a)(b)分别为DAB-DETR和DN-DETR的锚点与目标示例。每条箭头均从一个锚点出发
和 p 箭头指向目标。每个箭头的颜色显示其l_1长度，较冷的颜色表示较短的箭头。

为了衡量匹配结果，我们设计了如下指标。对于一张训练图像，我们将Transformer解码器在第i轮次预测的物体记为$\mathbf{O^i} = \{O_0^i, O_1^i, ..., O_{N-1}^i\}$，其中$N$为预测物体数量；将真实标注物体记为$\mathbf{T} = \{T_0, T_1, T_2, ..., T_{M-1}\}$，其中$M$为真实物体数量。经过二分匹配后，我们计算索引向量$\mathbf{V^i} = \{V_0^i, V_1^i, ..., V_{N-1}^i\}$来存储第$i$轮次的匹配结果如下。

$$V_n^i = \begin{cases} m, & \text{if } O_n^i \text{ matches } T_m \\ -1, & \text{if } O_n^i \text{ matches nothing} \end{cases} \qquad (1)$$

我们将训练图像在epoch i的不稳定性定义为其V^i与V^{i-1}之间的差值，计算公式为

$$IS^i = \sum_{j=0}^{N} \mathbb{1}(V_n^i \neq V_n^{i-1}) \qquad (2)$$

其中1(·)是指示函数。1(x) =当x为真时取值为1，否则为0。整个数据集在第i个epoch的不稳定性是通过对所有图像的不稳定性数值取平均得到的。为了公式简洁起见，在式(1)和式(2)中我们省略了图像的索引标记。

图3展示了我们的DN-DETR（去噪DETR）与DAB-DETR在IS上的对比。我们在COCO 2017验证集[13]上进行了此项评估，该数据集平均每幅图像包含7.36个目标。因此，最大的可能IS为7.36 × 2 = 14.72。图3清晰地表明，我们的方法有效缓解了匹配不稳定的问题。

3.2 使查询搜索更加本地化

我们还证明了DN-DETR能够通过缩小锚点与对应目标之间的距离来辅助检测。DETR[1]通过可视化展示其位置查询具有多种操作模式，这使得查询需从广阔区域中搜索预测框。而DN-DETR的初始锚点（位置查询）与目标之间的平均距离显著更小。如图4(a)所示，我们计算了DAB-DETR与我们的模型在最后解码层中初始锚点与匹配真实框之间的平均l_1距离。

由于去噪训练旨在让模型从接近真实标注的噪声框中进行重建，模型将在更局部的范围内进行预测搜索，这使得每个查询专注于附近区域，避免了查询之间潜在的预测冲突。图4(b)和(c)展示了DAB-DETR与DN-DETR中锚点与目标的部分示例。每条箭头从锚点出发，指向其匹配的真实标注框。我们通过颜色来反映箭头的长度。锚点与目标之间距离的缩短使得训练过程更为容易，从而收敛更快。

4 DN-DETR

4.1 概述

我们基于DAB-DETR[14]的架构来实现我们的训练方法。与DAB-DETR类似，我们明确地将解码器查询表述为边界框坐标。我们的架构与其唯一的不同之处在于解码器嵌入，这里被指定为类别标签嵌入以支持标签去噪。我们的主要贡献是如图6所示的训练方法。

与DETR类似，我们的架构包含一个Transformer编码器和一个Transformer解码器。在编码器端，图像特征通过CNN骨干网络提取，随后结合位置编码输入到Transformer编码器中，以获得精细化后的图像特征。在解码器端，查询向量被送入解码器，通过交叉注意力机制来搜索目标对象。

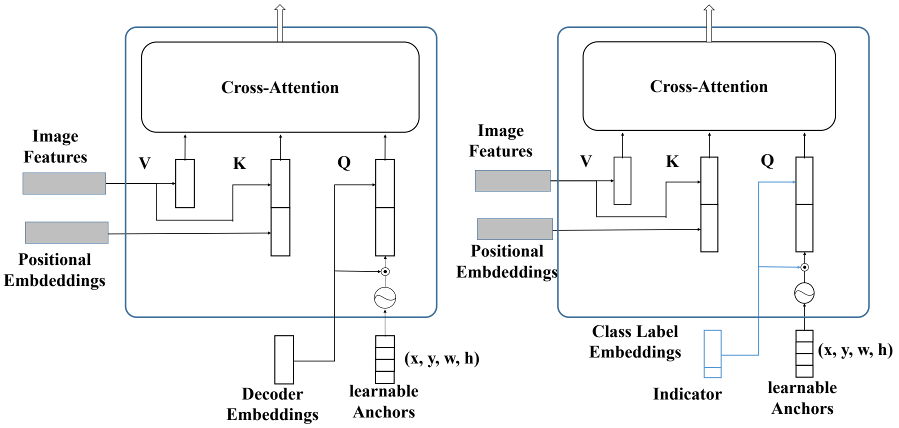
我们将解码器查询表示为$\mathbf{q} = \{q_0, q_1, ..., q_{N-1}\}$，Transformer解码器的输出为$\mathbf{o}$ =$\{o_0, o_1, ..., o_{N-1}\}$。同时，采用$F$和$A$分别表示经过Transformer编码器精炼后的图像特征，以及基于去噪任务设计得出的注意力掩码。我们的方法可表述如下。

$$\mathbf{o} = D(\mathbf{q}, F|A) \qquad (3)$$

其中D表示Transformer解码器。

解码器查询包含两个部分。一是匹配部分，其输入为可学习的锚点，处理方式与DETR相同，即采用二分图匹配策略，通过匹配解码器输出来逼近真实框-标签对。另一部分是去噪部分，其输入为添加噪声的真实（GT）框-标签对（后文统称为GT对象），该部分的输出旨在重构这些GT对象。

在下文中，我们滥用符号表示法，将去噪部分记为$\mathbf{q} = \{q_0, q_1, ..., q_{K-1}\}$，而匹配部分则



(a) Cross-attention in decoder of DAB-DETR

(b) Cross-attention in decoder of DN-DETR

Fig. 5. Comparison of the cross-attention part DAB-DETR and our DN-DETR (a)DAB-DETR directly uses dynamically updated anchor boxes to provide both a reference query point (x, y) and a reference anchor size (w, h) to improve the cross-attention computation. (b) DN-DETR specifies the decoder embeddings as label embeddings and adds an indicator to differentiate the denoising task and matching task.

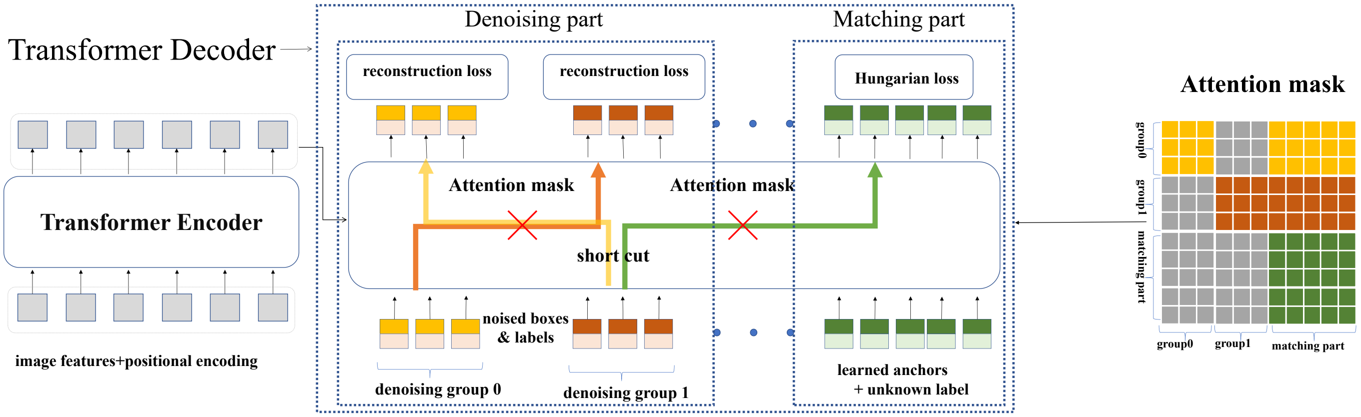


Fig. 6. The overview of our training method. There are two parts of queries, namely the denoising part and the matching part. The denoising part contains ≥ 1 denoising groups. The attention masks from the matching part to the denoising part and among denoising groups are set to 1 (block) to block information leakage. In the figure, the yellow, brown and green grids in the attention mask represent 0 (unblock) and grey grids represent 1 (block).

part as $\mathbf{Q} = \{Q_0, Q_1, ..., Q_{L-1}\}$. So the formulation of our method becomes

$$\mathbf{o} = D(\mathbf{q}, \mathbf{Q}, F|A) \tag{4}$$

To increase the denoising efficiency, we propose to use multiple versions of noised GT objects in the denoising part. Furthermore, we utilize an attention mask to prevent information leakage from the denoising part to the matching part and among different noised versions of the same GT object.

4.2 Intro to DAB-DETR

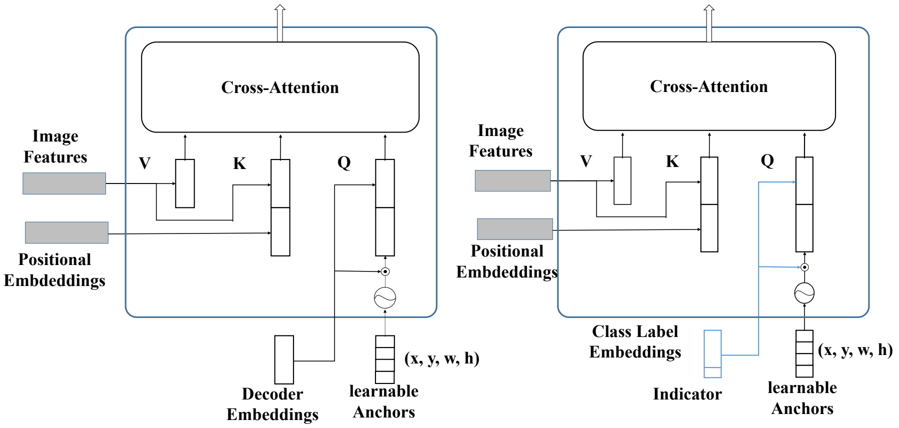
Many recent works associate DETR queries with different positional information. DAB-DETR follows this analysis and explicitly formulates each query as 4D anchor coordinates. As shown in Fig. 5(a), a query is specified as a tuple (x, y, w, h), where x, y are the center coordinates and w, h are the corresponding width and height of each box. In addition, the anchor coordinates are dynamically updated layer by layer. The output of each decoder layer contains a tuple $(\Delta x, \Delta y, \Delta w, \Delta h)$ and the anchor is updated to $(x + \Delta x, y + \Delta y, w + \Delta w, h + \Delta h)$.

Note that our proposed method is mainly a training method that can be integrated into any DETR-like model. To test on DAB-DETR, we only add minimal modifications: specifying the decoder embedding as label embedding, as shown in Fig. 5(b).

4.3 Denoising

For each image, we collect all GT objects and add random noises to both their bounding boxes and class labels. To maximize the utility of denoising learning, we use multiple noised versions for each GT object.

We consider adding noise to boxes in two ways: center shifting and box scaling. We define λ_1 and λ_2 as the noise scale of these 2 noises. 1) **center shifting:** we add a random noise $(\Delta x, \Delta y)$, to the box center and make sure that $|\Delta x| < \frac{\lambda_1 w}{2}$ and $|\Delta y| < \frac{\lambda_1 h}{2}$, where $\lambda_1 \in (0, 1)$ so that the center of the noised box will still lie inside the original bounding box. 2) **box scaling:** we set a hyper-parameter $\lambda_2 \in (0, 1)$. The width and height of the box are randomly sampled



(a) Cross-attention in decoder of DAB-DETR

(b) Cross-attention in decoder of DN-DETR

图5. DAB-DETR与我们的DN-DETR在交叉注意力部分的对比。(a)DAB-DETR直接使用动态更新的锚框，同时提供参考查询点(x,y)和参考锚尺寸(w,h)以改进交叉注意力计算。(b)DN-DETR将解码器嵌入指定为标签嵌入，并添加指示符以区分去噪任务和匹配任务。

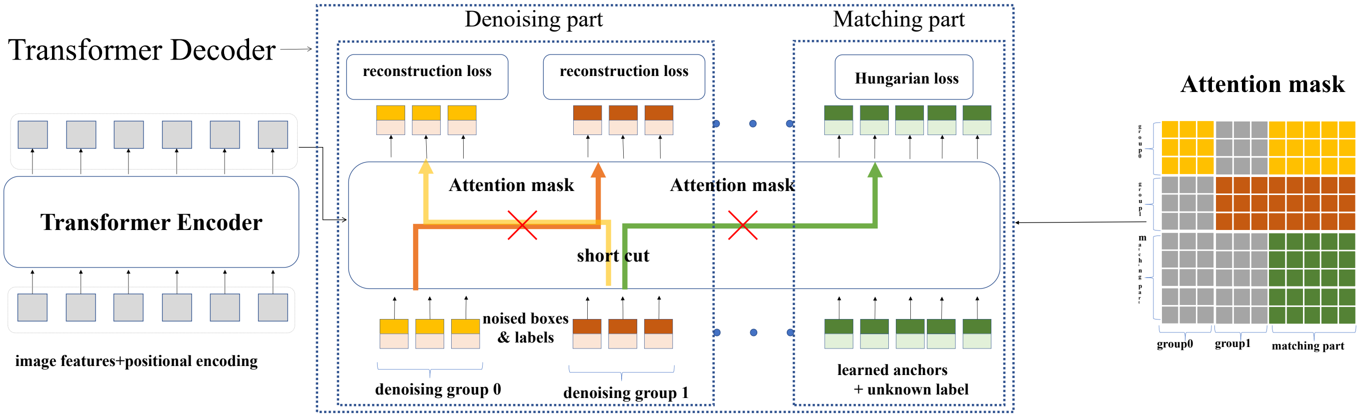


图6. 我们训练方法的概览。查询分为两部分，即去噪部分和匹配部分。去噪部分包含≥ 1个去噪组。从匹配部分到去噪部分以及去噪组之间的注意力掩码被设为1（阻断），以防止信息泄露。图中，注意力掩码的黄色、棕色和绿色网格代表0（未阻断），灰色网格代表1（阻断）。

部分为$\mathbf{Q}=\{Q_0,Q_1,...,Q_{L-1}\}$。因此，我们方法的公式化表达变为

$$\mathbf{o}=D(\mathbf{q},\mathbf{Q},F|A) \tag{4}$$

为了提高去噪效率，我们提出在去噪部分使用多个带噪GT对象版本。此外，我们采用注意力掩码机制，以防止信息从去噪部分泄露至匹配部分，以及同一GT对象不同带噪版本间的信息泄露。

4.2 DAB-DETR简介

许多近期研究将DETR查询与不同的位置信息相关联。DAB-DETR遵循这一分析思路，明确将每个查询表述为4D锚点坐标。如图5(a)所示，查询被定义为元组(x,y,w,h)，其中x,y表示中心坐标，w,h对应每个框的宽度和高度。此外，锚点坐标会逐层动态更新。每个解码器层的输出包含一个元组（{v10*}），锚点更新为（{v11*}）。

需要注意的是，我们提出的方法主要是一种训练方法，可以集成到任何类DETR模型中。为了在DAB-DETR上进行测试，我们仅做了最小限度的修改：如图5(b)所示，将解码器嵌入指定为标签嵌入{v*}。

4.3 去噪

对于每张图像，我们收集所有真实标注对象，并向它们的边界框和类别标签添加随机噪声。为了最大化去噪学习的效用，我们为每个真实标注对象生成多个噪声版本。

我们考虑以两种方式为边界框添加噪声：中心偏移和框缩放。定义λ_1和λ_2分别为这两种噪声的噪声尺度。1) 中心偏移：向框中心添加随机噪声$(\Delta x,\Delta y)$，并确保$|\Delta x|<\frac{\lambda_1 w}{2}$和$|\Delta y|<\frac{\lambda_1 h}{2}$，其中$\lambda_1\in(0,1)$，使得加噪后的框中心仍位于原始边界框内。2) 框缩放：设置一个超参数$\lambda_2\in($0,1)，框的宽度和高度将据此随机采样。

in $[(1-\lambda_2)w, (1+\lambda_2)w]$ and $[(1-\lambda_2)h, (1+\lambda_2)h]$, respectively.

For label noising, we adopt label flipping, which means we randomly flip some GT labels to other labels. Label flipping forces the model to predict the GT labels according to the noised boxes to better capture the label-box relationship. We have a hyper-parameter γ to control the ratio of labels to flip. The reconstruction losses are l_1 loss and GIOU loss for boxes and focal loss [12] for class labels as in DAB-DETR. We use a function $\delta(\cdot)$ to denote the noised GT objects. Therefore, each query in the denoising part can be represented as $q_k = \delta(t_m)$ where t_m is m-th GT object.

Notice that denoising is only considered in training, during inference the denoising part is removed, leaving only the matching part.

4.4 Attention Mask

Attention mask is a component of great importance in our model. Without an attention mask, the denoising training will compromise the performance instead of improving it as shown in Table 5.

To introduce an attention mask, we need first to divide the noised GT objects into groups. Each group is a noised version of all GT objects. The denoising part becomes

$$\mathbf{q} = \{\mathbf{g_0}, \mathbf{g_1}, ..., \mathbf{g_{P-1}}\} \tag{5}$$

where $\mathbf{g_p}$ is defined as the p-th **denoising group**. Each denoising group contains M queries where M is the number of GT objects in the image. So we have

$$\mathbf{g_p} = \{q_0^p, q_1^p, ..., q_{M-1}^p\} \tag{6}$$

where $q_m^p = \delta(t_m)$.

The purpose of the attention mask is to prevent information leakage. There are two types of potential information leakage. One is that the matching part may see the noised GT objects and easily predict GT objects. The other is that one noised version of a GT object may see another version. Therefore, our attention mask is to make sure the matching part cannot see the denoising part and the denoising groups cannot see each other as shown in Fig. 6.

We use $\mathbf{A} = [\mathbf{a}_{ij}]_{W\times W}$ to denote the attention mask where $W = P \times M + N$. P and M are the number of groups and GT objects. N is the number of queries in the matching part. We let the first $P \times M$ rows and columns represent the denoising part and the latter represents the matching part. $a_{ij} = 1$ means the i-th query cannot see the j-th query and $a_{ij} = 0$ otherwise. We devise the attention mask as follows

$$a_{ij} = \begin{cases} 1, & \text{if } j < P \times M \text{ and } \lfloor \frac{i}{M} \rfloor \neq \lfloor \frac{j}{M} \rfloor; \\ 1, & \text{if } j < P \times M \text{ and } i \geq P \times M; \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Note that whether the denoising part can see the matching part or not will not influence the performance, since the queries of the matching part are learned queries that contain no information about the GT objects.

The extra computation introduced by multiple denoising groups is negligible—when 5 denoising groups are introduced, GFLOPs for training are only increased from 94.4 to 94.6 for DAB-DETR with a ResNet-50 backbone, and there is no computation overhead for testing.

4.5 Label Embedding

The decoder embedding is specified as label embedding in our model to support both box denoising and label denoising. Except for the 80 classes in COCO 2017 [13], we also consider an unknown class embedding that is used in the matching part to be semantically consistent with the denoising part. We also append an indicator to label embedding. The indicator is 1 if a query belongs to the denoising part and 0 otherwise.

4.6 Compatibility with Deformable Attention Design

DN-Deformable-DETR: To show the effectiveness of denoising training applied in other attention designs, we also integrate denoising training into Deformable DETR as DN-Deformable-DETR. We follow the same setting as Deformable DETR but specify its query into 4D boxes as in DAB-DETR to better use denoising training. Note that this is our original deformable model in the conference version, in which we only add deformable attention to Transformer encoders.

When comparing in the standard 50 epoch setting, to eliminate any misleading information that the performance improvement of DN-Deformable-DETR may result from the explicit query formulation of anchor boxes, we also implement a strong baseline DAB-Defromable-DETR for comparison. It formulates the queries of Deformable DETR as anchor boxes without using denoising training, while all the other settings are the same.

DN-Deformable-DETR++: We further incorporate the deformable attention in our decoder and optimize our model to build DN-Deformable-DETR++, which converges much faster and improves the final results. We also follow DAB-Defromable-DETR to build a strong baseline DAB-Defromable-DETR++ to show our performance improvement in the ablations.

4.7 Introducing DN to Other DETR-like models with different anchor formulations

In the aforementioned sections, we build DN-DETR upon DAB-DETR [14] with explicit 4D anchor box formulation. As shown in Fig. 6, denoising is only a training method and can be plugged into other detection models to accelerate training. In this section, we will extend denoising training to other DETR-like models.

4.7.1 Introducing DN to Anchor DETR with 2D Anchors

We first demonstrate its effectiveness by adding it to Anchor DETR [21], which formulates positional queries as 2D anchor points. For DN-Anchor-DETR, though it can be easily modified to 4D anchors to achieve better results, we strictly follow Anchor DETR to add noise only to 2D anchors. A 2D anchor corresponds to the center point of a box. Hence we only use center shifting noise (described in Sec. 4.3). In this way, we plug in the denoising training task for anchor points without introducing other modifications.

4.7.2 Introducing DN to Vanilla DETR without Explicit Anchors

Vanilla DETR [1] differs from DAB-DETR in that its positional queries are high dimensional vectors without explicit

在$[(1-\lambda_2)w,(1+\lambda_2)w]$和$[(1-\lambda_2)h,(1+\lambda_2)h]$中，分别。

对于标签噪声处理，我们采用标签翻转策略，即随机将部分真实标签（GT）翻转为其他类别。标签翻转迫使模型根据带噪边界框预测真实标签，以更好地捕捉标签与边界框之间的关系。我们设定了一个超参数γ来控制翻转标签的比例。重构损失包括边界框的l_1损失和GIOU损失，以及类别标签的focal loss[12]，这与DAB-DETR中的设置一致。我们使用函数$\delta(\cdot)$表示加噪后的真实目标对象。因此，去噪部分的每个查询可表示为$q_k=\delta(t_m)$，其中t_m代表第m个真实目标对象。

需要注意的是，去噪仅在训练阶段被考虑，在推理过程中去噪部分会被移除，仅保留匹配部分。

4.4 注意力掩码

注意力掩码是我们模型中至关重要的组成部分。如表5所示，若缺少注意力掩码，去噪训练不仅无法提升性能，反而会损害模型表现。

为了引入注意力掩码，我们首先需要将带噪声的GT对象划分为若干组。每组都是所有GT对象的一个噪声版本。去噪部分变为

$$\mathbf{q}=\{\mathbf{g_0},\mathbf{g_1},...,\mathbf{g_{P-1}}\}\tag{5}$$

其中$\mathbf{g_p}$被定义为第p个去噪组。每个去噪组包含M个查询，而M表示图像中GT对象的数量。因此我们有

$$\mathbf{g_p}=\{q_0^p,q_1^p,...,q_{M-1}^p\}\tag{6}$$

其中$q_m^p=\delta(t_m)$。

注意力掩码的目的是防止信息泄露。存在两种潜在的信息泄露情况：一是匹配部分可能看到带噪声的真实对象，从而轻易预测出真实对象；二是某个真实对象的带噪声版本可能会看到另一个版本。因此，我们的注意力掩码确保匹配部分无法看到去噪部分，同时各去噪组之间也互不可见，如图6所示。

我们使用$\mathbf{A}\ =[\mathbf{a}_{ij}]_{W\times W}$表示注意力掩码，其中$W=P\times M+N$。$P$和$M$分别为组数和真实目标数量。$N$是匹配部分中的查询数量。我们让前$P\times M$行和列代表去噪部分，其余部分代表匹配部分。$a_{ij}$ =1表示第i个查询无法看到第j个查询，a_{ij} =0则表示可以。我们设计的注意力掩码如下

$$a_{ij}=\begin{cases}1, & \text{if } j<P\times M \text{ and } \lfloor\frac{i}{M}\rfloor\neq\lfloor\frac{j}{M}\rfloor;\\ 1, & \text{if } j<P\times M \text{ and } i\geq P\times M;\\ 0, & \text{otherwise.}\end{cases}\tag{7}$$

需要注意的是，去噪部分能否看到匹配部分不会影响性能，因为匹配部分的查询是学习得到的查询，不包含关于真实目标（GT objects）的任何信息。公式标记{v*}保持不变。

引入多个去噪组带来的额外计算量可以忽略不计——当使用5个去噪组时，基于ResNet-50骨干网络的DAB-DETR训练GFLOPs仅从94.4增加到94.6，且测试阶段不会产生任何计算开销。

4.5 标签嵌入

解码器嵌入在我们的模型中被指定为标签嵌入，以同时支持框去噪和标签去噪。除了COCO 2017数据集中的80个类别外，我们还引入了一个未知类别的嵌入表示，用于匹配部分，以确保与去噪部分在语义上保持一致。此外，我们在标签嵌入中附加了一个指示符：若查询属于去噪部分，则该指示符为1，否则为0。

4.6 与可变形注意力设计的兼容性

DN-Deformable-DETR：为了展示去噪训练在其他注意力设计中的有效性，我们还将去噪训练集成到Deformable DETR中，形成DN-Deformable-DETR。我们遵循与Deformable DETR相同的设置，但将其查询指定为4D边界框（如DAB-DETR所示），以更好地利用去噪训练。请注意，这是我们会议版本中的原始可变形模型，其中我们仅在Transformer编码器中添加了可变形注意力。

在标准的50轮训练设置下进行比较时，为消除DN-Deformable-DETR性能提升可能源于锚框显式查询表述的误导性信息，我们还实现了一个强基线DAB-Deformable-DETR作为对比。该模型将Deformable DETR的查询表述为锚框，但未使用去噪训练，其余所有设置均保持一致。

DN-Deformable-DETR**++**：我们进一步在解码器中整合了可变形注意力机制，并优化模型以构建DN-Deformable-DETR++，其收敛速度显著加快且最终结果更优。同时，我们遵循DAB-Deformable-DETR的架构建立了强基线模型DAB-Deformable-DETR++，用以在消融实验中验证我们的性能提升。

4.7 将DN引入其他采用不同锚点公式的类DETR模型

在前述章节中，我们在DAB-DETR[14]基础上构建了DN-DETR，采用了显式的4D锚框公式。如图6所示，去噪仅作为一种训练方法，可嵌入其他检测模型以加速训练过程。本节将把去噪训练推广至其他类DETR模型中。

4.7.1 Introducing DN to Anchor DETR with 2D Anchors

我们首先通过将其引入Anchor DETR[21]来验证其有效性，该方法将位置查询表述为二维锚点。对于DN-Anchor-DETR，虽然可以轻松修改为四维锚点以获得更好效果，但我们严格遵循Anchor DETR仅对二维锚点添加噪声。二维锚点对应框的中心点，因此我们仅采用中心偏移噪声（如第4.3节所述）。通过这种方式，我们在不引入其他改动的情况下，为锚点嵌入了去噪训练任务。

4.7.2 Introducing DN to Vanilla DETR without Explicit Anchors

Vanilla DETR [1] 与 DAB-DETR 的不同之处在于，其位置查询是高维向量，没有明确的

TABLE 1
Results for our DN-DETR and other detection models under the same setting. All DETR-like models except DETR use 300 queries, while DETR uses 100.

Model	#epochs	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$	GFLOPs	Params
DETR-R50 [1]	500	42.0	62.4	44.2	20.5	45.8	61.1	86	41M
Faster RCNN-FPN-R50 [18]	108	42.0	62.1	45.5	26.6	45.5	53.4	180	42M
Anchor DETR-R50 [21]	50	42.1	63.1	44.9	22.3	46.2	60.0	–	39M
Conditional DETR-R50 [15]	50	40.9	61.8	43.3	20.8	44.6	59.2	90	44M
DAB-DETR-R50 [14]	50	42.2	63.1	44.7	21.5	45.7	60.3	94	44M
DN-DETR-R50	50	**44.1(+1.9)**	64.4	46.7	22.9	48.0	63.4	94	44M
DETR-R101 [1]	500	43.5	63.8	46.4	21.9	48.0	61.8	152	60M
Faster RCNN-FPN-R101 [18]	108	44.0	63.9	47.8	27.2	48.1	56.0	246	60M
Anchor DETR-R101 [21]	50	43.5	64.3	46.6	23.2	47.7	61.4	–	58M
Conditional DETR-R101 [15]	50	42.8	63.7	46.0	21.7	46.6	60.9	156	63M
DAB-DETR-R101 [14]	50	43.5	63.9	46.6	23.6	47.3	61.5	174	63M
DN-DETR-R101	50	**45.2(+1.7)**	65.5	48.3	24.1	49.1	65.1	174	63M
DETR-DC5-R50 [1]	500	43.3	63.1	45.9	22.5	47.3	61.1	187	41M
Anchor DETR-DC5-R50 [21]	50	44.2	64.7	47.5	24.7	48.2	60.6	151	39M
Conditional DETR-DC5-R50 [15]	50	43.8	64.4	46.7	24.0	47.6	60.7	195	44M
DAB-DETR-DC5-R50 [14]	50	44.5	65.1	47.7	25.3	48.2	62.3	202	44M
DN-DETR-DC5-R50	50	**46.3(+1.8)**	66.4	49.7	26.7	50.0	64.3	202	44M
DETR-DC5-R101 [1]	500	44.9	64.7	47.7	23.7	49.5	62.3	253	60M
Anchor DETR-R101 [21]	50	45.1	65.7	48.8	25.8	49.4	61.6	–	58M
Conditional DETR-DC5-R101 [15]	50	45.0	65.5	48.4	26.1	48.9	62.8	262	63M
DAB-DETR-DC5-R101 [14]	50	45.8	65.9	49.3	27.0	49.8	63.8	282	63M
DN-DETR-DC5-R101	50	**47.3(+1.5)**	67.5	50.8	28.6	51.5	65.0	282	63M

meanings. For DN-Vanilla-DETR, we can simply use linear box embedding to embed noised boxes into the same dimension as DETR queries. The content query part is the same as DAB-DETR, and we use label embedding to embed labels into content queries. After obtaining content and position queries, following Vanilla DETR, we can add the label embedding and box embedding together as DETR queries.

4.8 Introducing DN to Faster R-CNN for Traditional Detectors

Apart from accelerating DETR-like models, denoising training can also be used to accelerate traditional CNN detectors. We take Faster R-CNN [18] as an example and add denoising training to it. The detection head of Faster R-CNN works in a similar way as the decoder of DETR-based models, where the major differences lie in 1) feature extraction: Faster R-CNN uses RoI pooling while DETR uses cross attention to extract features. and 2) label-assignment scheme: Faster R-CNN adopts a one-to-many label assignment (one GT object can be matched with multiple predicted objects), while DETR adopts a one-to-one label assignment (one GT object can only be matched with one predicted object). As the denoising part trains in parallel with the original matching part in detection models and is irrelevant to feature extraction schemes, denoising training can be easily applied to these traditional detectors.

Fundamentally, the idea of denoising training in DETR is to bypass the unstable label assignment and directly learn bounding box regression. Though Faster R-CNN does not have bipartite matching, it also has label assignment controlled by the IoU threshold. Therefore, denoising training can also serve as a shortcut to help learn bounding box regression without label assignment in traditional models. Therefore, we add noised boxes to the detection head of Faster R-CNN in parallel with the original boxes from the RPN. These noised boxes will directly regress the GT to improve training. Note that as Faster R-CNN does not have an initial content part, we only use box denoising training.

4.9 Introducing DN to Mask2Former for Segmentation Models

We also show the feasibility of adding denoising training to segmentation models such as Mask2Former [5]. Mask2Former adopts a DETR-like architecture and proposes masked attention to extract features for segmentation tasks. More specifically, each decoder layer predicts segmentation masks, which are passed to the subsequent decoder layer as the attention mask to pool features. Therefore, following the idea of denoising training in detection models, we can add noise to the GT masks and feed them to the decoder as the attention mask. The training objective of these noised masks is to directly predict the GT mask, which bypasses the bipartite match and serves as a shortcut to directly learn mask refinement.

To verify the effectiveness of denoising training on masks, we build a simple baseline by adding simple shifting noise to the mask. Without changing the shape or size of the mask, we shift the whole GT mask on the x-axis and y-axis by a random value, which is the same as the center shifting noise as described in Sec. 4.3. This simple baseline already demonstrates the effectiveness of denoising training.

表1 相同设置下我们的DN-DETR与其他检测模型的结果。除DETR外，所有类DETR模型均使用300个查询，而DETR使用100个。

Model	#epochs	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$	GFLOPs	Params
DETR-R50 [1]	500	42.0	62.4	44.2	20.5	45.8	61.1	86	41M
Faster RCNN-FPN-R50 [18]	108	42.0	62.1	45.5	26.6	45.5	53.4	180	42M
Anchor DETR-R50 [21]	50	42.1	63.1	44.9	22.3	46.2	60.0	–	39M
Conditional DETR-R50 [15]	50	40.9	61.8	43.3	20.8	44.6	59.2	90	44M
DAB-DETR-R50 [14]	50	42.2	63.1	44.7	21.5	45.7	60.3	94	44M
DN-DETR-R50	50	**44.1(+1.9)**	64.4	46.7	22.9	48.0	63.4	94	44M
DETR-R101 [1]	500	43.5	63.8	46.4	21.9	48.0	61.8	152	60M
Faster RCNN-FPN-R101 [18]	108	44.0	63.9	47.8	27.2	48.1	56.0	246	60M
Anchor DETR-R101 [21]	50	43.5	64.3	46.6	23.2	47.7	61.4	–	58M
Conditional DETR-R101 [15]	50	42.8	63.7	46.0	21.7	46.6	60.9	156	63M
DAB-DETR-R101 [14]	50	43.5	63.9	46.6	23.6	47.3	61.5	174	63M
DN-DETR-R101	50	**45.2(+1.7)**	65.5	48.3	24.1	49.1	65.1	174	63M
DETR-DC5-R50 [1]	500	43.3	63.1	45.9	22.5	47.3	61.1	187	41M
Anchor DETR-DC5-R50 [21]	50	44.2	64.7	47.5	24.7	48.2	60.6	151	39M
Conditional DETR-DC5-R50 [15]	50	43.8	64.4	46.7	24.0	47.6	60.7	195	44M
DAB-DETR-DC5-R50 [14]	50	44.5	65.1	47.7	25.3	48.2	62.3	202	44M
DN-DETR-DC5-R50	50	**46.3(+1.8)**	66.4	49.7	26.7	50.0	64.3	202	44M
DETR-DC5-R101 [1]	500	44.9	64.7	47.7	23.7	49.5	62.3	253	60M
Anchor DETR-R101 [21]	50	45.1	65.7	48.8	25.8	49.4	61.6	–	58M
Conditional DETR-DC5-R101 [15]	50	45.0	65.5	48.4	26.1	48.9	62.8	262	63M
DAB-DETR-DC5-R101 [14]	50	45.8	65.9	49.3	27.0	49.8	63.8	282	63M
DN-DETR-DC5-R101	50	**47.3(+1.5)**	67.5	50.8	28.6	51.5	65.0	282	63M

含义。对于DN-Vanilla-DETR，我们可以简单地使用线性框嵌入将带噪声的框嵌入到与DETR查询相同的维度。内容查询部分与DAB-DETR相同，我们使用标签嵌入将标签嵌入到内容查询中。在获得内容和位置查询后，遵循Vanilla DETR的做法，我们可以将标签嵌入和框嵌入相加，作为DETR查询。

4.8 为传统检测器引入DN到Faster R-CNN

除了加速类DETR模型，去噪训练同样可用于加速传统CNN检测器。我们以Faster R-CNN[18]为例，为其引入去噪训练。Faster R-CNN的检测头工作原理与基于DETR模型的解码器类似，主要差异在于：1）特征提取方式——Faster R-CNN采用RoI池化，而DETR使用交叉注意力机制；2）标签分配策略——Faster R-CNN采用一对多标签分配（一个真实目标可匹配多个预测目标），DETR则采用一对一标签分配（一个真实目标仅匹配一个预测目标）。由于去噪模块与检测模型中原始匹配模块并行训练，且与特征提取方案无关，去噪训练能轻松适配这些传统检测器。

从根本上说，DETR中的去噪训练理念是为了绕过不稳定的标签分配，直接学习边界框回归。尽管Faster R-CNN没有二分匹配机制，但它同样通过IoU阈值来控制标签分配。因此，去噪训练也能作为一种捷径，助力边界框的学习。

传统模型中无需标签分配的回归。因此，我们在Faster R-CNN的检测头部与来自RPN的原始框并行地添加了噪声框。这些噪声框将直接回归到GT（真实值）以优化训练。需要注意的是，由于Faster R-CNN最初不包含内容部分，我们仅采用框去噪训练。

4.9 为分割模型引入DN到Mask2Former

我们还展示了将去噪训练引入如Mask2Former[5]等分割模型的可行性。Mask2Former采用类DETR架构，提出掩码注意力机制以提取分割任务特征。具体而言，每个解码器层会预测分割掩码，这些掩码将作为注意力掩码传递至后续解码器层以汇聚特征。因此，借鉴检测模型中去噪训练的思路，我们可以向真实掩码(GT masks)添加噪声，并将其作为注意力掩码输入解码器。这些带噪掩码的训练目标是直接预测原始真实掩码，从而绕过二分图匹配过程，形成直接学习掩码优化的捷径。

为了验证掩码去噪训练的有效性，我们通过向掩码添加简单平移噪声构建了一个简易基线。在不改变掩码形状或大小的前提下，我们将整个真实掩码沿x轴和y轴随机平移一定数值，这与第4.3节描述的中心平移噪声方法一致。这一简单基线已充分证明了去噪训练的有效性。

TABLE 2
Results for our DN-DETR and other detection models on the 1x setting. Superscript † indicates that we check with the authors of Dynamic DETR through private communication, their encoder deign makes their single-scale and multi-scale results almost identical.

Model	MultiScale	#epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	GFLOPs	Params
Faster R-CNN-FPN-R50 1x [18]	✓	12	37.9	58.8	41.1	22.4	41.1	49.1	180	40M
DETR-R50 1x [1]		12	15.5	29.4	14.5	4.3	15.1	26.7	86	41M
DAB-DETR-DC5-R50 [14]		12	38.0	60.3	39.8	19.2	40.9	55.4	216	44M
DN-DETR-DC5-R50		12	**41.7(+3.7)**	61.4	44.1	21.2	45.0	60.2	216	44M
Deformable DETR-R50 1x [25]	✓	12	37.2	55.5	40.5	21.1	40.7	50.5	173	40M
Dynamic DETR-R50† 1x w/o dynamic encoder	✓	12	40.2	58.6	43.4	−−	−	−	−	−
Dynamic DETR-R50† 1x [6]	✓	12	42.9	61.0	46.3	24.6	44.9	54.4	−	−
DN-Deformable-DETR-R50	✓	12	**43.4**	61.9	47.2	24.8	46.8	59.4	195	48M
DN-Deformable-DETR-R50++	✓	12	**46.0**	63.8	49.9	27.7	49.1	62.3	−	47M
DAB-DETR-DC5-R101 [14]		12	40.3	62.6	42.7	22.2	44.0	57.3	282	63M
DN-DETR-DC5-R101		12	**42.8(+2.5)**	62.9	45.7	23.3	46.6	61.3	282	63M
Faster R101 FPN [18]	✓	108	44.0	63.9	47.8	27.2	48.1	56.0	246	60M
DN-Deformable-DETR-R101	✓	12	**44.1**	62.8	47.9	26.0	47.8	61.3	275	67M

TABLE 3
Extending denoisng training to other detection and segmentation models. Superscript * means this result is from the ablation experiments of the original paper that uses our denoising training.

Model	MultiScale	#epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	GFLOPs	Params
Extending DN to other detection models										
Anchor-DETR-DC5-R50 [21]		12	38.2	58.6	40.6	20.3	41.9	53.1	−	37M
DN-Anchor-DETR-DC5-R50		12	**39.4(+1.2)**	59.1	41.8	19.6	43.4	56.0	−	37M
Group-DAB-DETR-DC5-R50 [3]		12	41.9	−	−	23.3	45.6	58.4	−	−M
DN-Group-DAB-DETR-DC5-R50* [3]		12	**44.5(+2.6)**	−	−	25.9	48.2	62.2	−	−M
Faster R-CNN-FPN-R50 [21]	✓	12	37.9	58.8	41.1	22.4	41.1	49.1	180	40M
DN-Faster R-CNN-FPN-R50	✓	12	**38.4(+0.5)**	59.1	41.5	22.7	41.6	50.4	180	40M
SAM-DETR++-R50 [23]	✓	12	43.2	61.5	46.5	25.5	46.5	58.6	203	55M
DN-SAM-DETR++-R50* [23]	✓	12	**44.8(+1.6)**	62.6	47.9	26.7	48.2	60.9	203	55M
DINO-R50 w/o DN [24]	✓	12	46.0	64.0	49.9	29.3	49.2	60.5	279	47M
DINO-R50 w/ DN* [24]	✓	12	**47.4(+1.4)**	64.6	51.3	30.0	50.7	61.8	279	47M
Vanilla-DETR-R50 [1]		300	40.6	61.6	−	19.9	44.3	60.2	86	41M
DN-Vanilla-DETR-R50		300	**42.6(+2.0)**	62.3	44.9	21.6	46.1	61.4	86	37M
Extending DN to segmentation models										
Mask DINO-R50 w/o mask DN [11]	✓	12	40.7	62.8	43.7	21.0	43.4	60.6	234	50M
Mask DINO-R50 w/ mask DN * [11]	✓	12	**41.4(+0.7)**	62.9	44.6	21.1	44.2	61.4	234	50M
Mask2Former-R50 [5]	✓	12	38.7	59.8	41.2	18.2	41.5	59.8	226	44M
DN-Mask2Former-R50	✓	12	**39.7(+1.0)**	60.8	42.3	19.1	42.7	61.2	226	44M

5 Experiment

5.1 Setup

Dataset: We show the effectiveness of DN-DETR on the challenging MS-COCO 2017 [13] Detection task. MS-COCO is composed of 160K images with 80 categories. These images are divided into `train2017` with 118K images, `val2017` with 5K images, and `test2017` with 41K images. In all our experiments, we train the models on `train2017` and test on `val2017`. Following the common practice, we report the standard mean average precision (AP) result on the COCO validation dataset under different IoU thresholds and object scales.

Implementation Details: We test the effectiveness of the denoising training on DAB-DETR, which is composed of a CNN backbone, multiple Transformer encoder layers, and decoder layers. We also show that denoising training can be plugged into other DETR-like models to boost performance. For example, our DN-Deformable-DETR is built upon Deformable DETR in a multi-scale setting.

We adopt several ResNet models [9] pre-trained on ImageNet as our backbones and report our results on 4 ResNet settings: ResNet-50 (R50), ResNet-101 (R101), and their 16×-resolution extensions ResNet-50-DC5 (DC5-R50) and ResNet-101-DC5 (DC5-R101). For hyperparameters, we follow DAB-DETR to use a 6-layer Transformer encoder and a 6-layer Transformer decoder and 256 as the hidden dimension. We add uniform noise on boxes and set the hyperparameters with respect to noise as $\lambda_1 = 0.4$, $\lambda_2 = 0.4$, and $\gamma = 0.2$. For the learning rate scheduler, we use an initial learning rate (lr) 1×10^{-4} and drop lr at the 40-th epoch by multiplying 0.1 for the 50-epoch setting and at the 11-th epoch by multiplying 0.1 for the 12-epoch setting. We use the AdamW optimizer with weight decay of 1×10^{-4} and train our model on 8 Nvidia A100 GPUs. The batch size is 16. Unless otherwise specified, we use 5 denoising groups.

We conduct a series of experiments to demonstrate the performance improvement as shown in Table 1, where we follow the basic settings in DAB-DETR without any bells

表2 展示了我们的DN-DETR及其他检测模型在1x设置下的结果。上标†表示我们通过私下沟通与Dynamic DETR的作者确认，其编码器设计使得他们的单尺度和多尺度结果几乎相同。

Model	MultiScale	#epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	GFLOPs	Params
Faster R-CNN-FPN-R50 1x [18]	✓	12	37.9	58.8	41.1	22.4	41.1	49.1	180	40M
DETR-R50 1x [1]		12	15.5	29.4	14.5	4.3	15.1	26.7	86	41M
DAB-DETR-DC5-R50 [14]		12	38.0	60.3	39.8	19.2	40.9	55.4	216	44M
DN-DETR-DC5-R50		12	**41.7(+3.7)**	61.4	44.1	21.2	45.0	60.2	216	44M
Deformable DETR-R50 1x [25]	✓	12	37.2	55.5	40.5	21.1	40.7	50.5	173	40M
Dynamic DETR-R50† 1x w/o dynamic encoder	✓	12	40.2	58.6	43.4	−−	−	−	−	−
Dynamic DETR-R50† 1x [6]	✓	12	42.9	61.0	46.3	24.6	44.9	54.4	−	−
DN-Deformable-DETR-R50	✓	12	**43.4**	61.9	47.2	24.8	46.8	59.4	195	48M
DN-Deformable-DETR-R50++	✓	12	**46.0**	63.8	49.9	27.7	49.1	62.3	−	47M
DAB-DETR-DC5-R101 [14]		12	40.3	62.6	42.7	22.2	44.0	57.3	282	63M
DN-DETR-DC5-R101		12	**42.8(+2.5)**	62.9	45.7	23.3	46.6	61.3	282	63M
Faster R101 FPN [18]	✓	108	44.0	63.9	47.8	27.2	48.1	56.0	246	60M
DN-Deformable-DETR-R101	✓	12	**44.1**	62.8	47.9	26.0	47.8	61.3	275	67M

表3 将去噪训练扩展到其他检测和分割模型。上标*表示该结果来自原论文中使用我们提出的去噪训练进行的消融实验。

Model	MultiScale	#epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	GFLOPs	Params
Extending DN to other detection models										
Anchor-DETR-DC5-R50 [21]		12	38.2	58.6	40.6	20.3	41.9	53.1	–	37M
DN-Anchor-DETR-DC5-R50		12	**39.4(+1.2)**	59.1	41.8	19.6	43.4	56.0	–	37M
Group-DAB-DETR-DC5-R50 [3]		12	41.9	–	–	23.3	45.6	58.4	–	–M
DN-Group-DAB-DETR-DC5-R50* [3]		12	**44.5(+2.6)**	–	–	25.9	48.2	62.2	–	–M
Faster R-CNN-FPN-R50 [21]	✓	12	37.9	58.8	41.1	22.4	41.1	49.1	180	40M
DN-Faster R-CNN-FPN-R50	✓	12	**38.4(+0.5)**	59.1	41.5	22.7	41.6	50.4	180	40M
SAM-DETR++-R50 [23]	✓	12	43.2	61.5	46.5	25.5	46.5	58.6	203	55M
DN-SAM-DETR++-R50* [23]	✓	12	**44.8(+1.6)**	62.6	47.9	26.7	48.2	60.9	203	55M
DINO-R50 w/o DN [24]	✓	12	46.0	64.0	49.9	29.3	49.2	60.5	279	47M
DINO-R50 w/ DN* [24]	✓	12	**47.4(+1.4)**	64.6	51.3	30.0	50.7	61.8	279	47M
Vanilla-DETR-R50 [1]		300	40.6	61.6	–	19.9	44.3	60.2	86	41M
DN-Vanilla-DETR-R50		300	**42.6(+2.0)**	62.3	44.9	21.6	46.1	61.4	86	37M
Extending DN to segmentation models										
Mask DINO-R50 w/o mask DN [11]	✓	12	40.7	62.8	43.7	21.0	43.4	60.6	234	50M
Mask DINO-R50 w/ mask DN * [11]	✓	12	**41.4(+0.7)**	62.9	44.6	21.1	44.2	61.4	234	50M
Mask2Former-R50 [5]	✓	12	38.7	59.8	41.2	18.2	41.5	59.8	226	44M
DN-Mask2Former-R50	✓	12	**39.7(+1.0)**	60.8	42.3	19.1	42.7	61.2	226	44M

5 实验

5.1 设置

数据集：我们展示了DN-DETR在具有挑战性的MS-COCO 2017[13]检测任务上的有效性。MS-COCO包含16万张图像，涵盖80个类别。这些图像被划分为train2017（11.8万张）、val2017（5千张）和test2017（4.1万张）。在所有实验中，我们使用train2017训练模型，并在val2017上进行测试。遵循常规做法，我们报告了在不同IoU阈值和物体尺度下，COCO验证数据集上的标准平均精度（AP）结果。

实现细节：我们在DAB-DETR上测试了去噪训练的有效性，该模型由CNN骨干网络、多个Transformer编码器层和解码器层组成。我们还展示了去噪训练可以嵌入到其他类似DETR的模型中以提高性能。例如，我们的DN-Deformable-DETR是在多尺度设置下基于Deformable DETR构建的。

我们采用在ImageNet上预训练的多个ResNet模型[9]作为骨干网络，并在4种ResNet配置下报告结果：ResNet-50（R50）、ResNet-101（R101）及其16×分辨率扩展版本ResNet-50-DC5（DC5-R50）和ResNet-101-DC5（DC5-R101）。超参数设置方面，我们遵循DAB-DETR，使用6层Transformer编码器与6层Transformer解码器，隐藏维度设为256。我们在边界框上添加均匀噪声，并将噪声相关超参数设为λ_1 =0.4、λ_2 =0.4和γ =0.2。学习率调度器采用初始学习率（lr）1×10^{-4}，在50轮训练设置中第40轮时乘以0.1进行衰减，在12轮设置中则于第11轮进行相同衰减。优化器选用AdamW，权重衰减为1×10^{-4}，模型在8块NVIDIA A100 GPU上训练，批量大小为16。除非特别说明，默认使用5组去噪。

我们进行了一系列实验以展示性能提升，如表1所示，其中我们遵循DAB-DETR的基本设置，未添加任何额外修饰。

TABLE 4
Best results for our DN-DETR and other detection models with the ResNet-50 backbone. * indicates it is the test-dev result.

Model	MultiScale	#epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	GFLOPs	Params
Deformable DETR-R50 [25]	✓	50	43.8	62.6	47.7	26.4	47.1	58.0	173	40M
SMCA-R50 [8]	✓	50	43.7	63.6	47.2	24.2	47.0	60.4	152	40M
TSP-RCNN-R50 [19]	✓	96	45.0	64.5	49.6	29.7	47.7	58.0	188	–
Dynamic DETR-R50* [6]	✓	50	47.2	65.9	51.1	28.6	49.3	59.1	–	–
DAB-Deformable-DETR-R50	✓	50	46.9	66.0	50.8	30.1	50.4	62.5	195	48M
DN-Deformable-DETR-R50	✓	50	**48.6**	67.4	52.7	31.0	52.0	63.7	195	48M
DN-Deformable-DETR-R50++	✓	50	**49.5**	67.6	53.8	31.3	52.6	65.4	–	47M

and whistles in training. To compare with the state-of-the-art performance in the 12 epoch setting (the so-called 1× setting in Detectron2) and the standard 50 epoch setting (most widely used in DETR-like models) in Table 2 and 4, we follow DAB-DETR to use 3 pattern embeddings as in Anchor DETR [21]. All our comparisons with DAB-DETR and its variants are under exactly the same setting.

DN-Deformable-DETR and DN-Deformable-DETR++: For DN-Deformable-DETR with only deformable encoder, we use 10 denoising groups. For DN-Deformable-DETR++ with deformable attention in both encoder and decoder, we use 5 denoising groups. Note that we strictly follow Deformable DETR to use multi-scale (4 scale) features without FPN. Dynamic DETR [6] adds FPN and more scales (5 scales) which can further boost the performance, but our performance still exceeds theirs.

Faster R-CNN and Anchor DETR: We use 10 and 5 denoising groups respectively.

DINO: To test the effectiveness of denoising training in DINO, we only use our proposed DN without its proposed contrastive DN and keep all the other components in DINO. We use 5 denoising groups.

Mask DINO: Mask DINO incorporates both box denoising and mask denoising. To show performance improvement over segmentation tasks, we keep the box denoising part and only remove the mask denoising to study its effectiveness. We use 5 denoising groups under this setting.

Mask2Former: Mask2Former is only designed for segmentation tasks. Therefore, we only add mask denoising training in our experiments. We use 5 denoising groups under this setting.

Our proposed denoising training has been incorporated into many subsequent works and also implemented in detrex (https://github.com/IDEA-Research/detrex).

5.2 Denoising Training Improves Performance

To show the absolute performance improvement compared with DAB-DETR and other single-scale DETR models, we conduct a series of experiments using different backbones under the basic single-scale settings. The results are summarized in Table 1.

The results show that we achieve the best results among single-scale models with all four commonly used backbones. For example, compared with our baseline DAB-DETR under exactly the same setting, we achieve **+1.9** AP absolute improvement with ResNet-50. The table also shows that denoising training adds negligible parameters and computation.

5.3 1× Setting

With denoising training, the detection task can be accelerated by a large margin. As shown in Table 2, we compare our method with both a traditional detector [18] and some DETR-like models, including DETR [1], Dynamic DETR [6], and Deformable DETR [25]. Note that Dynamic DETR [6] adopts a dynamic encoder, for a fair comparison, we also compare with its version without a dynamic encoder.

Under the same setting with the DC5-R50 backbone, DN-DETR can outperform DAB-DETR by **+3.7** AP within 12 epochs. Compared with other models, DN-Deformable-DETR achieves the best results in the 12 epoch setting. It is worth noting that our DN-Deformable-DETR achieves **44.1** AP within 12 epochs with the ResNet-101 backbone, which surpasses Faster R-CNN ResNet-101 trained for 108 epochs (9× faster).

5.4 Extending DN to Other Detection and Segmentation Models

To further validate the effectiveness of denoising training, we extend this method to other detection and segmentation model, as shown in Table 3. The experimental results indicate that denoising training is a universal training method to boost performance.

For example, we improve the DETR-like detection models significantly by $1.2 - 2.6$ AP under the 12-epoch setting. The results also reveal that

- Denoising training is compatible with other positional query formulations, for example, Vallina DETR with high dimensional vectors, Anchor DETR with 2D anchor points, and DAB-DETR with 4D anchor boxes.
- Our method is only a training method and also compatible with other methods, for example, deformable attention [25], semantic-alignment [23], and query selection[24], etc.

5.5 Compared with State-of-Art Detectors

We also conduct experiments to compare our method with multi-scale models. The results is summarized in Table 4. Our proposed DN-Deformable-DETR achieves the best result **48.6** AP with the ResNet-50 backbone. To eliminate the performance improvement from formulating the queries of deformable DETR as anchor boxes, we further use a strong baseline DAB-Deformable-DETR without denoising

表4 采用ResNet-50骨干网络的DN-DETR及其他检测模型的最佳结果。*表示该结果为测试开发集结果。

Model	MultiScale	#epochs	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$	GFLOPs	Params
Deformable DETR-R50 [25]	✓	50	43.8	62.6	47.7	26.4	47.1	58.0	173	40M
SMCA-R50 [8]	✓	50	43.7	63.6	47.2	24.2	47.0	60.4	152	40M
TSP-RCNN-R50 [19]	✓	96	45.0	64.5	49.6	29.7	47.7	58.0	188	–
Dynamic DETR-R50* [6]	✓	50	47.2	65.9	51.1	28.6	49.3	59.1	–	–
DAB-Deformable-DETR-R50	✓	50	46.9	66.0	50.8	30.1	50.4	62.5	195	48M
DN-Deformable-DETR-R50	✓	50	**48.6**	67.4	52.7	31.0	52.0	63.7	195	48M
DN-Deformable-DETR-R50++	✓	50	**49.5**	67.6	53.8	31.3	52.6	65.4	–	47M

以及在训练中的技巧。为了与表2和表4中12个epoch设置（即Detectron2中所谓的1×设置）和标准的50个epoch设置（在类DETR模型中最广泛使用）的先进性能进行比较，我们遵循DAB-DETR的做法，采用Anchor DETR[21]中的3个模式嵌入。我们与DAB-DETR及其变体的所有比较均在完全相同的设置下进行。

DN-Deformable-DETR与DN-Deformable-DETR++：对于仅含可变形编码器的DN-Deformable-DETR，我们采用10组去噪任务。而在编码器和解码器均配备可变形注意力的DN-Deformable-DETR++中，则使用5组去噪任务。需注意的是，我们严格遵循Deformable DETR的做法，采用多尺度（4尺度）特征且不使用FPN。Dynamic DETR[6]通过添加FPN及更多尺度（5尺度）进一步提升了性能，但我们的方法仍超越其表现。

Faster R-CNN与Anchor DETR：我们分别使用了10组和5组去噪组。

DINO：为了验证DINO中降噪训练的有效性，我们仅采用提出的DN模块（不包含其对比式降噪部分），并保留DINO所有其他组件不变。实验中设置了5个降噪组。

Mask DINO：Mask DINO同时结合了框去噪与掩码去噪。为展示其在分割任务上的性能提升，我们保留了框去噪部分，仅移除掩码去噪以研究其有效性。在此设置下，我们采用了5组去噪操作。

Mask2Former：Mask2Former仅针对分割任务设计。因此，我们在实验中仅增加了掩模去噪训练。在此设置下，我们使用了5个去噪组。

我们提出的去噪训练方法已被众多后续研究采纳，并在detrex（https://github.com/IDEA-Research/detrex）中得到了实现。

5.2 去噪训练提升性能

为了展示与DAB-DETR及其他单尺度DETR模型相比的绝对性能提升，我们在基础单尺度设置下采用不同骨干网络进行了一系列实验。结果汇总于表1。

结果表明，在采用所有四种常用骨干网络的单尺度模型中，我们取得了最佳效果。例如，在与基线DAB-DETR完全相同的设置下对比，使用ResNet-50时我们实现了+1.9 AP的绝对提升。表格数据还显示，去噪训练增加的参数量和计算成本微乎其微。

5.3 1× 设置

通过去噪训练，检测任务的速度可以得到大幅提升。如表2所示，我们将本方法与传统的检测器[18]以及几种类DETR模型进行了对比，包括DETR[1]、Dynamic DETR[6]和Deformable DETR[25]。需要注意的是，Dynamic DETR[6]采用了动态编码器，为了公平比较，我们也对比了其未使用动态编码器的版本。

在同样采用DC5-R50骨干网络的设置下，DN-DETR能在12个epoch内以+3.7 AP的优势超越DAB-DETR。与其他模型相比，DN-Deformable-DETR在12个epoch的设置下取得了最佳结果。值得注意的是，我们的DN-Deformable-DETR在ResNet-101骨干网络下仅用12个epoch就达到了44.1 AP，超越了训练108个epoch的Faster R-CNN ResNet-101（提速×9倍）。

5.4 将DN扩展到其他检测与分割模型

为进一步验证去噪训练的有效性，我们将该方法扩展到其他检测与分割模型，如表3所示。实验结果表明，去噪训练是一种通用的训练方法，能够有效提升模型性能。

例如，在12个训练周期的设置下，我们将类DETR检测模型的性能显著提升了1.2 – 2.6 AP。结果还表明，

- 去噪训练与其他位置查询方案兼容，例如使用高维向量的Vanilla DETR、采用2D锚点的Anchor DETR，以及基于4D锚框的DAB-DETR。

- 我们的方法仅作为一种训练手段，同时兼容其他多种方法，例如可变形注意力{v*}[25]、语义对齐{v*}[23]以及查询选择{v*}[24]等。

5.5 与最先进的检测器对比

我们还进行了实验，将我们的方法与多尺度模型进行比较。结果总结在表4中。我们提出的DN-Deformable-DETR在使用ResNet-50骨干网络时取得了最佳结果48.6 AP。为了排除将可变形DETR查询表述为锚框带来的性能提升，我们进一步采用了不带去噪功能的强基线DAB-Deformable-DETR进行验证。

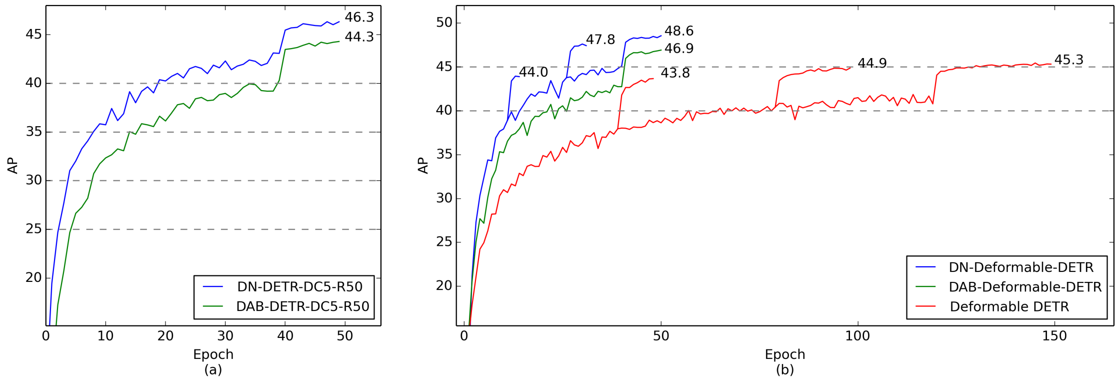


Fig. 7. (a) Convergence curves of DAB-DETR and DN-DETR with ResNet-DC5-50. Before learning rate drop, DN-DETR achieves 40 AP in 20 epochs, while DAB-DETR needs 40 epochs. (b) Convergence curves of multi-scale models with ResNet-50. With learning rate drop, DN-Deformable-DETR achieves 47.8 AP in 30 epochs, which is 0.9 AP higher than the converged DAB-Deformable-DETR.

TABLE 5
Ablation results for DN-DETR. All models are trained with the ResNet-50 backbone using 1 denoising group under the same default settings.

Box Denoising	Label Denoising	Attention Mask	AP
✓	✓	✓	43.4
✓		✓	43.0
		✓	42.2
✓	✓		24.0

TABLE 6
Ablation results for DN-DETR using different numbers of denoising groups. All models are trained with the ResNet-50 backbone under the same default setting.

	No Group	1 Group	5 Groups
R50	42.2	43.4	44.1
R50-DC5	44.5	45.6	46.3
R101	43.5	45.0	45.2
R101-DC5	45.8	46.5	47.3

training. The results show that we can still yield 1.7 AP absolute improvement. The performance improvement of DN-Deformable-DETR also indicates that denoising training can be integrated into other DETR-like models and improve their performance. Though it is not a fair comparison with Dynamic DETR as it includes a dynamic encoder and more scales (5 scales) with FPN, we still yield $+1.4$ AP improvement.

We also show the convergence curve in both single-scale and multi-scale settings in Fig. 7, where we drop the learning rate by 0.1 in multiple epochs in Fig. 7(b).

5.6 Ablation Study

5.6.1 *Effectiveness of each component*

We conduct a series of ablation studies with the ResNet-50 backbone trained for 50 epochs to verify the effectiveness of each component and report the results in Table 5 and Table 6. The results in Table 5 show that each component in denoising training contributes to performance improvement. Notably, without an attention mask to prevent information leakage, the performance degenerates significantly.

5.6.2 *Effectiveness of using more denoising groups*

We also analyze the influence of the number of denoising groups in our model, as shown in Table 6. The results indicate that adding more denoising groups improves performance, but the performance improvement becomes marginal as the number of denoising groups increases. Therefore, in our experiment, our default setting uses 5 denoising groups, but more denoising groups can further boost performance as well as faster convergence.

In Fig. 8, We explore the influence of noise scale. We run 20 epochs with batch size 64 and ResNet-50 backbone without learning rate drop. The results show that both center shifting and box scaling improve performance. But when the noise is too large, the performance drops.

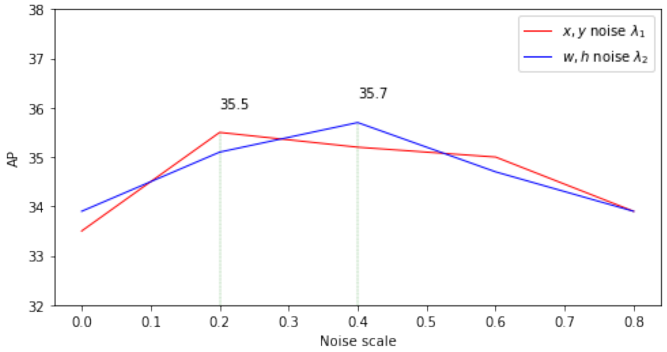


Fig. 8. DN-DETR in different noise scales. We fix one noise scale to 0.4 and change the other. Noise scale is defined in 4.3

5.6.3 *Acceleration Analysis*

We show how much our method can speed up training exactly in Table 1. Our method achieves results comparable to the baseline with only half of the training epochs, resulting in 2x acceleration.

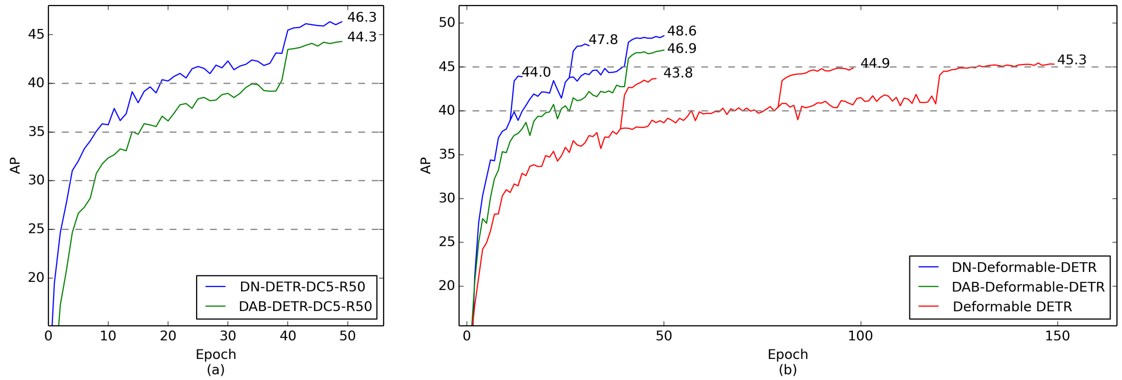


图7. (a) DAB-DETR与DN-DETR在ResNet-DC5-50架构下的收敛曲线。在学习率下降前，DN-DETR仅需20个epoch即可达到40 AP，而DAB-DETR需要40个epoch。(b) 基于ResNet-50的多尺度模型收敛曲线。伴随学习率下降，DN-Deformable-DETR在30个epoch内实现47.8 AP，较收敛后的DAB-Deformable-DETR高出0.9 AP。

表5 DN-DETR的消融实验结果。所有模型均采用ResNet-50骨干网络，在相同默认设置下使用1个去噪组进行训练。

Box Denoising	Label Denoising	Attention Mask	AP
✓	✓	✓	43.4
✓		✓	43.0
		✓	42.2
✓	✓		24.0

表6 使用不同去噪组数的DN-DETR消融实验结果。所有模型均在相同默认设置下使用ResNet-50骨干网络进行训练。

	No Group	1 Group	5 Groups
R50	42.2	43.4	44.1
R50-DC5	44.5	45.6	46.3
R101	43.5	45.0	45.2
R101-DC5	45.8	46.5	47.3

训练。结果表明，我们仍能实现1.7 AP的绝对提升。DN-Deformable-DETR的性能提升也表明，去噪训练可以整合到其他类DETR模型中并提高其性能。尽管与Dynamic DETR的对比并不完全公平，因为后者包含动态编码器和更多尺度（5个尺度）及FPN结构，但我们依然取得了+1.4 AP的改进。

我们还在图7中展示了单尺度和多尺度设置下的收敛曲线，其中在图7(b)的多轮训练中，我们将学习率降低了0.1。

5.6 消融研究

5.6.1 Effectiveness of each component

我们以ResNet-50为骨干网络进行了50轮训练的一系列消融实验，以验证各组件的有效性，并将结果记录在表5和表6中。表5的结果表明，去噪训练中的每个组件都对性能提升有所贡献。值得注意的是，如果没有注意力掩码来防止信息泄露，性能会显著下降。

5.6.2 Effectiveness of using more denoising groups

我们还分析了模型中降噪组数量的影响，如表6所示。结果表明，增加更多的降噪组能提升性能，但随着降噪组数量的增加，性能提升逐渐趋于平缓。因此，在我们的实验中，默认设置为使用5个降噪组，但更多的降噪组不仅能进一步提升性能，还能加快收敛速度。

在图8中，我们探究了噪声尺度的影响。我们以64的批量大小运行20个周期，采用ResNet-50主干网络且不降低学习率。结果表明，中心偏移和框缩放均能提升性能。但当噪声过大时，性能会下降。

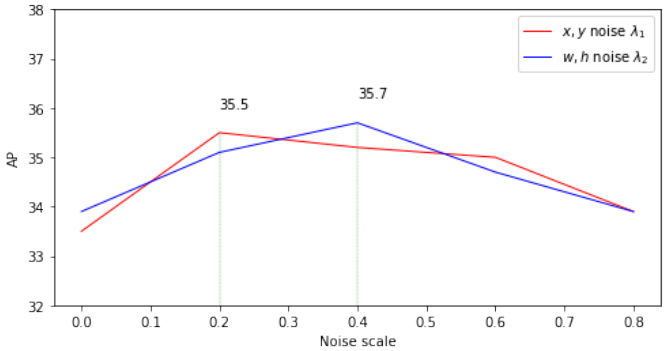


图8. 不同噪声尺度下的DN-DETR。我们将一个噪声尺度固定为0.4，并改变另一个。噪声尺度的定义见4.3节

5.6.3 Acceleration Analysis

我们通过表1精确展示了该方法能多大程度上加速训练。仅用一半的训练周期，我们的方法就取得了与基线相当的结果，实现了2倍的加速。

TABLE 1
Results of our method trained for 25 epochs and our baseline method trained for 50 epochs under the same settings. The results show we achieve 2x acceleration with denoising training.

Model	MultiScale	#epochs	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$	GFLOPs	Params
DAB-DETR-DC5-R50		50	44.5	65.1	47.7	25.3	48.2	62.3	202	44M
DN-DETR-DC5-R50		25	44.4	64.5	47.3	24.4	48.0	63.0	202	44M
DAB-Deformable-DETR-R50	✓	50	46.9	66.0	50.8	30.1	50.4	62.5	195	48M
DN-Deformable-DETR-R50	✓	25	46.8	65.5	50.8	28.9	50.2	62.5	195	48M
DAB-Deformable-DETR-R50++	✓	50	48.7	67.2	53.0	31.4	51.6	63.9	–	47M
DN-Deformable-DETR-R50++	✓	25	48.4	66.6	52.7	30.0	51.7	64.4	–	47M
Vanilla-DETR-R50 [1]		500	42.0	62.4	44.2	20.5	45.8	61.1	86	41M
DN-Vanilla-DETR-R50		250	42.2	61.8	44.6	20.5	46.0	61.3	86	37M

TABLE 2
We adopted five denoising groups for DN-DAB-DETR. The results are tested on the same GPUs for a fair comparison.

Model	Total Training time (min)	Training GFLOPs
DAB-DETR-R50	2555(50 epochs)	94.4
DN-DAB-DETR-R50	1443(25 epochs)	94.5

5.6.4 The training wall clock time and GFLOPs

We tested the training wall clock time and GFLOPs with 8 NVIDIA A100 GPUs as shown in Table 2. The total training time is calculated by multiplying the number of training epochs and the training time for each epoch. The training time per epoch is 51.1min and 57.7min for DAB-DETR-R50 and DN-DAB-DETR-R50, respectively. While denoising training introduces a minor training cost increase, it only needs about half the number of training epochs (25 epochs) to achieve the same performance as DAB-DETR-R50. The practical training speedup is indeed remarkable.

5.7 Other tasks and future work

5.7.1 Other Tasks

In addition to regular detection, our design of queries as anchor box + label makes the detection model capable of handling other tasks. For example, known object detection and known label detection. Note that the results shown in this section are just a preliminary exploration and not based on our well-trained model with the best hyper-parameters.

Known Object Detection: Assume we know a part of the objects in an image and want to predict the remaining objects. We want the known objects to help predict the unknown objects through co-occurrence relations. We did some preliminary exploration. We randomly divide the 80 classes of MS COCO2017 into 2 parts, including known classes and unknown classes. We put objects of known classes in the denoising part and want the matching part to predict the objects of the unknown classes. We do not use an attention mask so that the matching part can get useful information from the denoising part. Our experimental results are shown in Table 3. Compared with the evaluation without known boxes, the evaluation of the known object improves the performance, which indicates that co-occurrence helps the prediction of unknown boxes. Moreover, our DN-DETR trained with known objects exceeds DAB-DETR only trained on unknown classes when evaluating without known objects. This means the denoising of extra boxes from extra (known) classes also helps the performance of the unknown objects.

TABLE 3
Extra label prediction on COCO. We split the annotation of COCO class into known/unknown classes, where objects of known classes only appear in denoising part, and we evaluate the performance on the unknown classes. Cond means the result is evaluated with known objects.

Method	Setting	AP	AP(Cond)
DAB-DETR	0.7/0.3	38.4	-
DN-DETR	0.7/0.3	42.1	42.9
DAB-DETR	0.5/0.5	37.8	-
DN-DETR	0.5/0.5	39.1	40.3

Known Label Detection: For each image, we assume we know all the class labels in the image without box information. Since our model has interpreted the query embedding into class label embedding, we can seamlessly utilize these known labels to detect the boxes of each class label. For each class c in the image, we concatenate its label embedding with the indicator 1, which denotes a known label. We feed the concatenated vector into the decoder and let the decoder output all boxes of class c. To compare with methods without known labels and detect all objects in an image, we concatenate outputs of all classes and evaluate the result as shown in Table 4. By finetuning with known labels, the detection performance can be improved in only one epoch. Within 10 epochs of finetuning on pre-trained DN-DETR, the known label detection performance is improved to 46.6. This result demonstrates that given labels can significantly improve the detection performance.

5.7.2 Future Work

There are three potential future works to be mentioned here. One is zero-shot detection, and the other is progressive inference.

Zero-shot or Open Set Detection: Since we have decoupled decoder queries as anchor boxes and class labels, pre-trained class label embeddings can be fed into the class label part of the queries. To enable zero-shot detection, one can take

表1 在相同设置下，我们的方法训练25个周期与基线方法训练50个周期的结果对比。结果表明，通过去噪训练我们实现了2倍的加速效果。

Model	MultiScale	#epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	GFLOPs	Params
DAB-DETR-DC5-R50		50	44.5	65.1	47.7	25.3	48.2	62.3	202	44M
DN-DETR-DC5-R50		25	44.4	64.5	47.3	24.4	48.0	63.0	202	44M
DAB-Deformable-DETR-R50	✓	50	46.9	66.0	50.8	30.1	50.4	62.5	195	48M
DN-Deformable-DETR-R50	✓	25	46.8	65.5	50.8	28.9	50.2	62.5	195	48M
DAB-Deformable-DETR-R50++	✓	50	48.7	67.2	53.0	31.4	51.6	63.9	–	47M
DN-Deformable-DETR-R50++	✓	25	48.4	66.6	52.7	30.0	51.7	64.4	–	47M
Vanilla-DETR-R50 [1]		500	42.0	62.4	44.2	20.5	45.8	61.1	86	41M
DN-Vanilla-DETR-R50		250	42.2	61.8	44.6	20.5	46.0	61.3	86	37M

表2 我们为DN-DAB-DETR采用了五组去噪设置。所有结果均在相同GPU环境下测试，以确保对比的公平性。

Model	Total Training time (min)	Training GFLOPs
DAB-DETR-R50	2555(50 epochs)	94.4
DN-DAB-DETR-R50	1443(25 epochs)	94.5

5.6.4 *The training wall clock time and GFLOPs*

我们在8块NVIDIA A100 GPU上测试了训练实际耗时与GFLOPs，结果如表2所示。总训练时间通过训练轮次与每轮训练时长相乘得出。DAB-DETR-R50和DN-DAB-DETR-R50的每轮训练时长分别为51.1分钟和57.7分钟。虽然去噪训练会略微增加训练成本，但其仅需约一半的训练轮次（25轮）即可达到与DAB-DETR-R50相同的性能。实际训练加速效果确实显著。

5.7 其他任务与未来工作

5.7.1 *Other Tasks*

除了常规检测外，我们将查询设计为锚框+标签的做法使检测模型能够处理其他任务。例如已知物体检测与已知标签检测。需注意本节展示的结果仅为初步探索，并非基于我们经过充分训练且采用最优超参数的模型。

已知物体检测：假设我们已知图像中的部分物体，并希望预测剩余物体。我们期望通过共现关系，利用已知物体来辅助预测未知物体。我们进行了一些初步探索：将MS COCO2017的80个类别随机划分为已知类别和未知类别两部分。在去噪部分放置已知类别的物体，并希望匹配部分能预测未知类别的物体。我们没有使用注意力掩码，以便匹配部分能从去噪部分获取有用信息。实验结果如表3所示。与未使用已知框的评估相比，引入已知物体的评估提升了性能，这表明共现关系有助于未知框的预测。

此外，在评估时不包含已知对象的情况下，我们使用已知对象训练的DN-DETR超越了仅针对未知类别训练的DAB-DETR。这表明，通过额外（已知）类别对多余框的去噪处理，同样有助于提升未知对象的检测性能。

表3 COCO上的额外标签预测。我们将COCO类别的标注划分为已知/未知类别，其中已知类别的对象仅出现在去噪部分，而我们在未知类别上评估性能。Cond表示结果是在已知对象条件下评估的。

Method	Setting	AP	AP(Cond)
DAB-DETR	0.7/0.3	38.4	-
DN-DETR	0.7/0.3	42.1	42.9
DAB-DETR	0.5/0.5	37.8	-
DN-DETR	0.5/0.5	39.1	40.3

已知标签检测：对于每张图像，我们假设已知图像中所有类别的标签，但不包含边界框信息。由于我们的模型已将查询嵌入解析为类别标签嵌入，因此可以无缝利用这些已知标签来检测每个类别标签对应的边界框。对于图像中的每个类别c，我们将其标签嵌入与指示符1（表示已知标签）进行拼接，并将拼接后的向量输入解码器，使解码器输出类别c的所有边界框。为了与不使用已知标签的方法进行对比并检测图像中的所有对象，我们将所有类别的输出结果拼接后进行评测，如表4所示。通过使用已知标签进行微调，仅需一个训练周期即可提升检测性能。在预训练DN-DETR基础上进行10个周期的微调后，已知标签检测性能提升至46.6。这一结果表明，给定标签能显著提高检测性能。

5.7.2 *Future Work*

这里有三项潜在的未来工作值得提及。一是零样本检测，另一项是渐进式推理。

零样本或开放集检测：由于我们将解码器查询解耦为锚框和类别标签，预训练的类别标签嵌入可以直接输入到查询的类别标签部分。为了实现零样本检测，可以采取

TABLE 4
Known label detection results under ResNet-50 with 1 denoising group. 1ep and 10ep means finetuned 1 or 10 epochs from pretrained DN-DETR.

Method	Setting	AP
DAB-DETR	no knwon labels	42.2
DN-DETR	no knwon labels	43.4
DN-DETR	known label (1ep)	43.8
DN-DETR	known label (10ep)	46.6

80 classes of MSCOCO as phrases and collect phrase embeddings from a pre-trained language model as the class label embedding. With the pre-trained label embedding, it is possible to train a given class detector that takes a class label embedding as input and detects objects of the given classes. In inference time, class label embeddings from unseen classes can be fed into the decoder to achieve zero-shot detection.

Progressive inference: Based on known object detection, a progressive inference method can be designed. For example, we can train a DN-DETR capable of doing known object detection. In inference time, we let the detector predict objects, and then, we can choose the objects with the highest score and treat them as known objects to do known object detection. For each step of prediction, we choose objects with the highest score and add them to the known box set. After repeating for many times, we get the final prediction.

Classification before detection: As shown in Table 4, given labels can significantly improve the detection performance. Therefore, one potential future work is to add a muli-label classification network to provide labels and feed them to DN-DETR, which may help improve detection performance.

6 Conclusion

In this paper, we have analyzed the reason for the slow convergence of DETR training lying in the unstable bipartite matching and proposed a novel denoising training method to address this problem. Based on this analysis, we proposed DN-DETR by integrating denoising training into DAB-DETR to test its effectiveness. DN-DETR specifies the decoder embedding as label embedding and introduces denoising training for both boxes and labels. We also added denoising training to Deformable DETR to show its generality. The results show that denoising training significantly accelerates convergence and improves performance, leading to the best results in the 1x (12 epochs) setting with both ResNet-50 and ResNet-101 as the backbone. This study shows that denoising training can be easily integrated into DETR-like models as a general training method with only a small training cost overhead and bring in a remarkable improvement in terms of both training convergence and detection performance.

Limitations: In this work, the added noises are simply sampled from a uniform distribution. We have not explored more complex noising schemes and leave these for future work. Reconstructing noised data achieves great success in unsupervised learning and diffusion models. This work is an initial step to apply it to object detection. In the future, we will explore how to pre-train detectors on weakly labeled data with unsupervised learning techniques and explore applying other denoising training schemes in detection models.

References

[1] 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] 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.

[3] Qiang Chen, Xiaokang Chen, Gang Zeng, and Jingdong Wang. Group DETR: Fast Training Convergence with Decoupled One-to-Many Label Assignment. *arXiv preprint arXiv:2207.13085*, 2022.

[4] Ting Chen, Saurabh Saxena, Lala Li, David J. Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection, 2021.

[5] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022.

[6] 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*, pages 2988–2997, 2021.

[7] Enrico Maria Fenoaltea, Izat B Baybusinov, Jianyang Zhao, Lei Zhou, and Yi-Cheng Zhang. The Stable Marriage Problem: An interdisciplinary review from the physicist's perspective. *Physics Reports*, 2021.

[8] 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.

[9] 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.

[10] 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 Conference on Computer Vision and Pattern Recognition*, pages 13619–13627, 2022.

[11] Feng Li, Hao Zhang, Huaizhe xu, Shilong Liu, Lei Zhang, Lionel M. Ni, and Heung-Yeung Shum. Mask DINO: Towards A Unified Transformer-based Framework for Object Detection and Segmentation, 2022.

[12] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection, 2018.

[13] 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.

[14] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. DAB-DETR: Dynamic anchor boxes are better queries for DETR. In *International Conference on Learning Representations*, 2022.

[15] 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.

[16] Joseph Redmon and Ali Farhadi. YOLO9000: Better, Faster, Stronger, 2016.

[17] Joseph Redmon and Ali Farhadi. YOLOv3: An Incremental Improvement, 2018.

[18] 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.

[19] Zhiqing Sun, Shengcao Cao, Yiming Yang, and Kris Kitani. Rethinking transformer-based set prediction for object detection. *arXiv preprint arXiv:2011.10881*, 2020.

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin.

表4 ResNet-50下使用1个去噪组的已知标签检测结果。1ep和10ep表示从预训练的DN-DETR模型微调1或10个周期。

Method	Setting	AP
DAB-DETR	no knwon labels	42.2
DN-DETR	no knwon labels	43.4
DN-DETR	known label (1ep)	43.8
DN-DETR	known label (10ep)	46.6

将MSCOCO的80个类别作为短语，并从预训练的语言模型中收集短语嵌入作为类别标签的嵌入表示。借助这些预训练的标签嵌入，可以训练一个给定的类别检测器，该检测器以类别标签嵌入为输入，检测指定类别的物体。在推理阶段，将未见类别的标签嵌入输入解码器，即可实现零样本检测。

渐进式推理：基于已知目标检测，可以设计一种渐进式推理方法。例如，我们可以训练一个能够执行已知目标检测的DN-DETR模型。在推理阶段，先让检测器预测目标，随后选取置信度最高的目标作为已知对象进行检测。每一步预测时，都将当前得分最高的目标加入已知框集合中。经过多次迭代后，即可得到最终预测结果。

分类先于检测：如表4所示，给定标签能显著提升检测性能。因此，一项潜在的未来工作是增设一个多标签分类网络来提供标签，并将其输入DN-DETR，这可能有助于提升检测表现。

6 结论

本文分析了DETR训练收敛缓慢的原因在于不稳定的二分匹配，并提出了一种新颖的去噪训练方法以解决此问题。基于此分析，我们将去噪训练整合至DAB-DETR中，提出了DN-DETR以验证其有效性。DN-DETR将解码器嵌入指定为标签嵌入，并引入了针对边界框和标签的双重去噪训练。我们还为Deformable DETR添加了去噪训练以展示其通用性。实验结果表明，去噪训练显著加速了收敛速度并提升了性能，在使用ResNet-50和ResNet-101作为骨干网络的1x（12周期）训练设置下均取得了最佳结果。本研究表明，去噪训练能够以极小的训练成本开销作为通用训练方法，轻松集成到类DETR模型中，并在训练收敛速度和检测性能方面带来显著提升。

局限性：在本研究中，所添加的噪声仅从均匀分布中简单采样。我们尚未探索更复杂的噪声生成方案，这些将留待未来工作。在无监督学习和扩散模型中，对加噪数据的重建已取得显著成功。本研究将其应用于目标检测的初步尝试。未来，我们将探索如何利用无监督学习技术在弱标注数据上预训练检测器，并研究在检测模型中应用其他去噪训练方案。

参考文献

[1] 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 Conference onComputer Vision*，第213–229页。Springer出版社，2020年。 [2] 陈凯，庞江淼，王佳琪，熊宇，李晓晓，孙书阳，冯万森，刘子维，石建萍，欧阳万里等。面向实例分割的混合任务级联方法。载于 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*，第4974–4983页，2019年。 [3] 陈强，陈晓康，曾刚，王京东。Group DETR：通过解耦一对多标签分配实现快速训练收敛。*arXiv preprint arXiv:2207.13085*，2022年。 [4] 陈挺，Saurabh Saxena，李拉拉，David J. Fleet，Geoffrey Hinton。Pix2seq：一种用于目标检测的语言建模框架，2021年。 [5] 程博文，Ishan Misra，Alexander G Schwing，Alexander Kirillov，Rohit Girdhar。通用图像分割的掩蔽注意力掩码Transformer。载于*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*，第1290–1299页，2022年。 [6] 戴曦阳，陈寅鹏，杨建伟，张鹏川，袁璐，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/CVFInternational Conference on Computer Vision*，第2988–2997页，2021年。 [7] Enrico Maria Fenoaltea，Izat B Baybusinov，赵建阳，周磊，张毅成。稳定婚姻问题：物理学家的跨学科综述。*PhysicsReports*，2021年。 [8] 高鹏，郑明航，王晓刚，戴继峰，李洪生。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。 [9] 何恺明，张翔宇，任少卿，孙剑。深度残差学习在图像识别中的应用。载于*2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*，第770–778页，2016年。 [10] 李峰，张浩，刘世龙，郭健，倪力昂，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*，第13619–13627页，2022年。 [11] 李峰，张浩，徐怀哲，刘世龙，张磊，倪力昂，沈向洋。Mask DINO：迈向基于Transformer的目标检测与分割统一框架，2022年。 [12] 林钲贻，Priya Goyal，Ross Girshick，何恺明，Piotr Dollár。密集目标检测的焦点损失，2018年。 [13] 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出版社，2014年。 [14] 刘世龙，李峰，张浩，杨晓，齐先彪，苏航，朱军，张磊。DAB-DETR：动态锚框作为DETR的更优查询。载于*International Conference on Learning Representations*，2022年。 [15] 孟德普，陈晓康，范泽佳，曾刚，李厚强，袁宇辉，孙磊，王京东。条件式DETR实现快速训练收敛。*arXiv preprint arXiv:2108.06152*，2021年。 [16] Joseph Redmon，Ali Farhadi。YOLO9000：更好、更快、更强，2016年。 [17] Joseph Redmon，Ali Farhadi。YOLOv3：渐进式改进，2018年。 [18] 任少卿，何恺明，Ross Girshick，孙剑。Faster R-CNN：基于区域提议网络的实时目标检测。*IEEE Transactions on Pattern Analysis and Machine Intel-ligence*，39(6):1137–1149，2017年。 [19] 孙志青，曹盛操，杨一鸣，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。 [20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, Illia Polosukhin。

Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[21] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor DETR: Query design for transformer-based detector. *arXiv preprint arXiv:2109.07107*, 2021.

[22] 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.

[23] Gongjie Zhang, Zhipeng Luo, Yingchen Yu, Jiaxing Huang, Kaiwen Cui, Shijian Lu, and Eric P Xing. Semantic-Aligned Matching for Enhanced DETR Convergence and Multi-Scale Feature Fusion. *arXiv preprint arXiv:2207.14172*, 2022.

[24] 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, 2022.

[25] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

注意力就是你所需要的一切。在*Advances in neural information processing systems*，第5998–6008页，2017年。[21] 王英明、张翔宇、杨桐、孙剑。Anchor DETR：基于Transformer检测器的查询设计。*arXiv preprint arXiv:2109.07107*，2021年。[22] 姚竹雨、艾江波、李伯勋、张弛。Efficient DETR：利用密集先验改进端到端目标检测器。*arXiv preprint arXiv:2104.01318*，2021年。[23] 张功杰、罗志鹏、余英晨、黄嘉兴、崔凯文、陆世健、Eric P Xing。语义对齐匹配促进DETR收敛与多尺度特征融合。*arXiv preprint arXiv:2207.14172*，2022年。[24] 张浩、李峰、刘世龙、张磊、苏航、朱军、倪明选、沈向洋。DINO：带改进去噪锚框的DETR端到端目标检测，2022年。[25] 朱曦洲、苏伟杰、卢乐为、李斌、王晓刚、代季峰。Deformable DETR：可变形Transformer端到端目标检测。*arXiv preprint arXiv:2010.04159*，2020年。