DN-DETR: Accelerate DETR Training by Introducing Query DeNoising

摘要

本文提出了一种新的去噪训练方法 DN-DETR，旨在加速 DETR（DEtection TRansformer）的训练过程。研究者发现 DETR 训练收敛慢的问题源于二分图匹配的不稳定性，这在训练的早期阶段尤为明显。为了解决这个问题，DN-DETR 在 DETR 的基础上引入了去噪任务，通过在 Transformer 解码器中添加带有噪声的真实边界框（GT bounding boxes），训练模型重建原始框。这种方法有效地降低了二分图匹配的难度，从而加快了收敛速度。DN-DETR 的方法是通用的，可以通过添加少量代码轻松集成到任何 DETR-like 方法中。实验结果表明，DN-DETR 在相同的设置下实现了显著的性能提升，并且在较短的训练周期内达到了与基线相当的性能。

关键词：目标检测；DETR；去噪训练

1 引言

在计算机视觉领域，目标检测是一个核心任务，它涉及到在图像中识别和定位多个对象。随着深度学习技术的发展，基于卷积神经网络（CNN）的目标检测方法，如 Faster R-CNN，已经取得了显著的性能提升。然而，这些方法通常依赖于手工设计的锚点和非最大抑制（NMS）等后处理步骤，这些步骤限制了模型的端到端优化能力。

DETR 的出现打破了这一局限，它利用 Transformer 架构来处理图像，通过可学习的查询（queries）来探测图像特征，并使用二分图匹配来进行集合框预测。这种方法消除了对手工设计锚点的依赖，并且通过端到端的训练优化，提高了模型的灵活性和泛化能力。然而，DETR 的训练过程面临着收敛速度慢的问题，通常需要 500 个训练周期才能获得良好性能，这种慢收敛的问题在训练的早期阶段尤为明显，很大程度上限制了其在实际应用中的效率。

在这种背景下，研究者们开始探索如何改进 DETR 的训练过程，以提高其收敛速度。这不仅对提高目标检测任务的性能至关重要，也对推动整个计算机视觉领域的发展具有重要意义。研究者通过对 DETR 训练过程的分析，发现二分图匹配的不稳定性是导致训练收敛慢的主要原因。在训练的早期阶段，由于随机优化过程，同一个查询在不同周期内可能会与不同的真实边界框（GT boxes）匹配，这种不一致性使得优化目标变得模糊，从而阻碍了模型的快速学习。因此，提出一种新的训练方法来加速 DETR 的训练，成为了一个迫切需要解决的问题。

DN-DETR 正是在这样的背景下应运而生，它通过引入去噪训练，有效地稳定二分图匹配过程，减少训练过程中的不稳定性，从而有效地解决了 DETR 训练过程中的收敛速度问

题，为 DETR 及其变体模型的训练提供了一种新的优化策略。这种优化策略不仅能够显著提高 DETR 的训练效率，还能够在保持相同性能的同时减少所需的训练周期，这对于实际应用中的模型部署和迭代更新具有重要意义。此外，DN-DETR 的通用性意味着它可以轻松地集成到其他 DETR-like 模型中，为更广泛的目标检测和分割任务提供加速训练的解决方案。

2 相关工作

这部分内容主要介绍传统 CNN 检测器、基于 DETR 检测器的相关研究工作和基本原理。通过分析现有文献和研究成果，发现问题和解决问题，同时为新研究提供理论支持。

2.1 传统 CNN 检测器

现代目标检测模型大多依赖于卷积神经网络（CNN），近年来也取得了显著的成就。这些模型主要分为两类：一阶段检测器和两阶段检测器。两阶段检测器，如 HTC 和 Fast R-CNN，首先生成一系列区域建议，然后判断这些区域是否包含目标物体，并进行边界框回归以获得更精确的边界框。这种方法通常能提供更精确的边界框，但计算成本较高，速度较慢。Ren 等人 [7] 提出了一种端到端的方法，该方法利用区域建议网络来预测锚框。与两阶段方法不同，一阶段检测器，如 YOLO 900 和 YOLOv 3，在单次前向传播中同时执行目标分类和边界框定位，在特征图上预测边界框和类别概率，不需要区域建议或后续的边界框精炼步骤，直接预测真实边界框相对于锚框的偏移量。尽管这些一阶段检测器在多个数据集上展现出了卓越的性能，但它们对锚框生成策略非常敏感，并且依赖于一些手工设计的技术，例如非最大抑制（NMS）和标签分配规则。这些因素限制了它们的性能，并阻碍了模型的端到端优化。

2.2 基于 DETR 的检测器

2020 年，Carion 等 [1] 提出了一种名为 DETR（DEtection TRansformer）的新型目标检测器，该检测器完全基于 Transformer 架构，摒弃了传统的锚点机制。尽管 DETR 在性能上与 Faster-RCNN 相媲美，但其训练过程却因收敛速度缓慢而面临挑战，通常需要经过 500 个训练周期才能实现优异的性能。为了应对这一挑战，研究人员们采取了多种策略来加速 DETR 的训练。例如，Dai 等人 [3] 提出了一种动态解码器，该解码器能够在不同粒度上集中关注关键区域，从而简化了学习过程。Sun 等人 [8] 则采取了更为激进的方法，他们完全移除了 Transformer 解码器，提出了一种仅依赖编码器的 DETR 版本。此外，研究人员还在改进解码器查询方面进行了一系列工作。在 DETR 的训练过程中，有一系列研究致力于改进解码器的查询机制。Zhu 等人 [10] 开发了一个注意力模块，该模块仅关注参考点周围的特定采样点。Meng 等人 [6] 则将解码器的每个查询分解为内容和位置两个部分，并在交叉注意力计算中仅利用这两部分之间的相互作用。Yao 等人 [9] 利用区域建议网络（RPN）来提出最重要的 K 个锚点。DAB-DETR [5] 使用 4-D 边界框坐标作为查询，并采用逐层级联的方式更新这些框。尽管这些工作取得了一定的进展，但没有人将二分图匹配过程中的匈牙利损失视为训练收敛缓慢的关键因素。Sun 等人 [8] 通过使用预训练的 DETR 作为教师模型来指导学生模型的标签分配，发现这种分配策略在训练初期有助于收敛，但对最终性能影响不大。因此，他们认为匈牙利损失不是导致收敛缓慢的主要因素。然而，本研究提供了一种新的分析视角和有效的解决方案，得出了与之前不同的结论。

2.3 基于 DAB-DETR 检测框架评估训练策略

在本研究中，基于 DAB-DETR 这一基础检测架构来评估训练策略。为了支持标签去噪，本文方法采用带有附加指示符的标签嵌入来替代解码器的嵌入部分。所采取的方法与其他现有方法的主要区别在于训练策略。除了传统的匈牙利损失外，方法还引入了一个去噪损失，这个辅助任务相对简单，有助于加速训练过程并显著提升模型性能。Chen 等人 [2] 在其工作中通过添加合成噪声对象来增强序列，但这与本文方法有本质的不同。他们将噪声对象的目标类别设置为“噪声”，即不属于任何真实类别，这样做的目的是为了推迟句子结束（EOS）标记的出现，从而提高模型的召回率。相反，本文的目标是将噪声框恢复到其原始的真实框，这样做的目的是绕过二分图匹配，直接学习如何更准确地预测真实框的位置。

3 本文方法

3.1 本文方法概述

本文提出了一种新的训练方法，通过引入查询去噪任务来帮助稳定训练过程中的二部图匹配。具体实现为将带噪的 ground truth（GT）边界框作为带噪查询与可学习的锚查询一起输入到 Transformer 解码器。这两种查询具有相同的（x，y，w，h）输入格式，可以同时输入到 Transformer 解码器中。对于带噪查询，通过执行一个去噪任务来重建它们对应的 GT boxes。对于其他可学习的锚查询，使用与 vanilla DETR [1] 中相同的训练损失，包括二分匹配。由于带噪边界框不需要经过二分图匹配组件，因此去噪任务可以被视为一个更简单的辅助任务，帮助 DETR 缓解不稳定的离散二分图匹配，降低了二分图匹配的难度，并更快速地学习边界框预测，从而加速了训练过程的收敛。此部分的方法示意图如图 1所示：

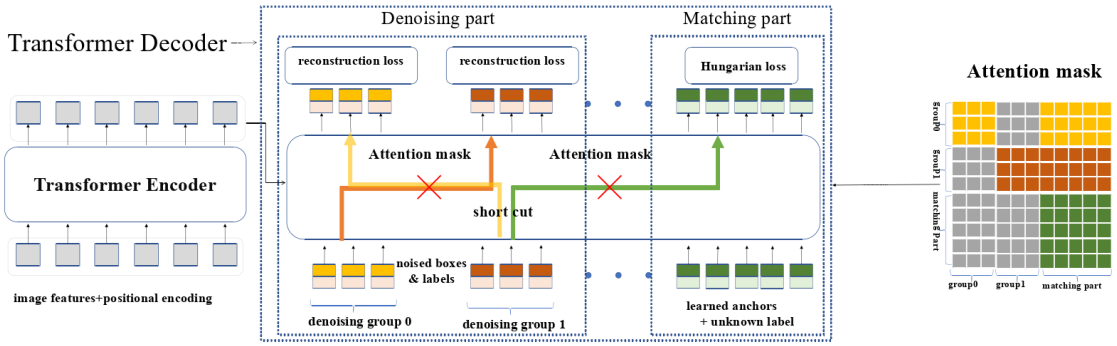

图 1. DN-DETR 训练方法示意图

本文的训练方法基于 DAB-DETR 的架构 [5] 来实现的。将解码器查询显式地表示为框坐标，但本文架构唯一的区别在于解码器嵌入，它被指定为类标签嵌入以支持标签去噪。与 DETR 类似，DN-DETR 的模型结构包含一个 Transformer 编码器和一个 Transformer 解码器。在编码器端，使用 CNN 主干提取图像特征，然后将图像特征输入到带有位置编码的 Transformer 编码器中，得到细化的图像特征。在解码器端，查询被输入到解码器中以通过交叉注意力搜索对象。编码器部分分为去噪部分（Denoising Part）和匹配部分（Matching Part），匹配部分的输入是可学习的锚，它们的处理方式与 DETR 相同，使用可学习的锚点进行二分图匹配，学习如何近似真实框和标签对，并学习用匹配的解码器输出来近似 GT box-label

pairs；去噪部分的输入是带噪的 GT box-label pairs，即添加了噪声的真实边界框和类别标签，包含 1 个去噪组。去噪部分的输出旨在重建 GT 对象，即训练模型从这些噪声版本中恢复出原始的 GT 对象，包括两个步骤：中心偏移和边界框缩放。

图 1 最右侧的图像是注意力掩模（Attention Mask），其中黄色、棕色和绿色网格代表 0 (未阻塞)，灰色网格代表 1 (阻塞)。通过引入注意力掩码，阻止信息泄露。这个掩码确保匹配部分不能看到去噪部分，并且去噪组之间也不能相互看到。

3.2 交叉注意力机制

最近的许多研究工作都关注于将 DETR 的查询与特定的空间位置信息关联起来。DAB-DETR 遵循这一趋势，明确地将每个查询表示为 4D 锚点坐标 $(x\ y\ w\ h)$。

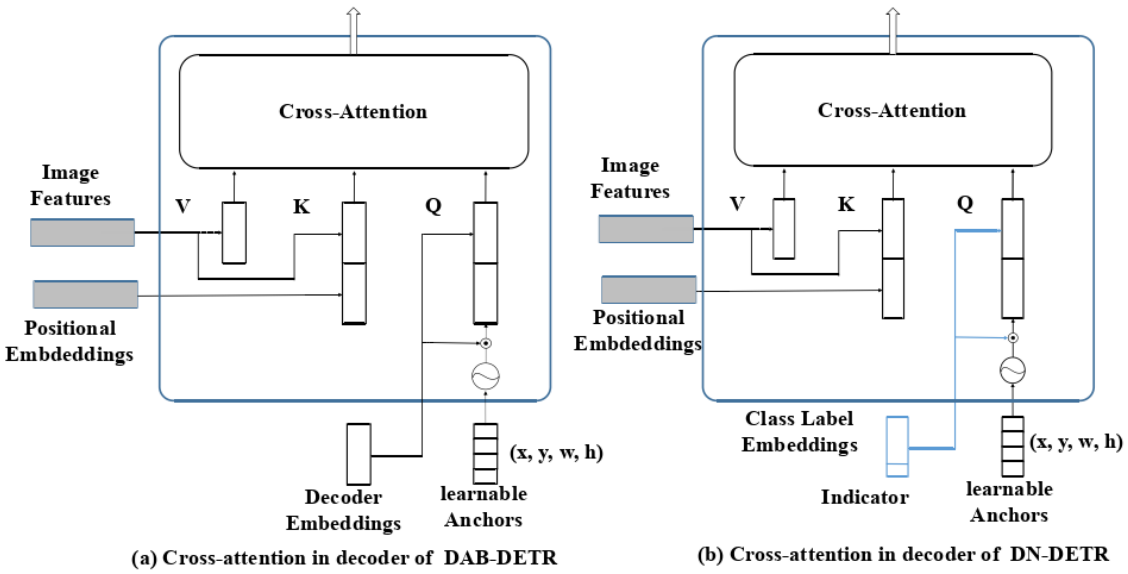

图 2. DAB-DETR 和 DN-DETR 的交叉注意力部分比较

如图 2（a）所示，每个查询被定义为一个包含中心点坐标（x，y）和框的宽度（w）和高度（h）的元组。此外，这些锚点坐标在网络的每一层都会被动态更新。在每个解码器层的输出中，都会包含一个更新后的元组 (x', y', w', h')，并且锚点会根据这些更新后的坐标进行调整得到 $(x+x', y+y', w+w', h+h')$。（b）所示部分则对 Decoder Embedding 进行了小修改，将解码器嵌入指定为标签嵌入，以同时支持 box 去噪和标签去噪。除了 COCO 2017 中的 80 个类之外，还考虑了一个未知的类嵌入，它用于匹配部分，以在语义上与去噪部分一致。此外，还添加了一个指示符（Indicator）来标记嵌入以区分去噪任务和匹配任务。如果查询属于去噪部分，则该指标为 1，否则为 0。

3.3 注意力掩码

注意力掩码的主要目的是确保模型在训练过程中，去噪部分（denoising part）不能看到匹配部分（matching part）的输出，以及去噪部分中的不同噪声版本之间也不能相互看到。这个掩码会阻止某些注意力权重的计算，从而防止信息泄露。这样可以避免模型直接学习到噪声版本和真实目标之间的直接映射，而是通过学习去噪任务来间接地学习目标检测。为了实

现这个目标，将噪声化的 GT 对象分成多个组，每个组包含所有 GT 对象的一个噪声版本。然后为每个去噪组定义了一个注意力掩码。

注意力掩码 A 是一个 W×W 的矩阵，其中 W 是去噪组的数量乘以 GT 对象的数量（P×M）加上匹配部分的查询数量（N）。掩码中的每个元素 aij 定义如下：

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

其中 $aij = 1$ 表示第 i 个查询不能看到第 j 个查询，其他情况下，$aij = 0$。通过引入注意力掩码，DN-DETR 能够更有效地进行去噪训练，加速模型的收敛，并提高目标检测的性能。

3.4 去噪加速训练策略

匈牙利匹配是图匹配中的一种流行算法。给定一个代价矩阵，算法输出一个最优的匹配结果。DETR 是第一个在目标检测中采用匈牙利匹配的算法，用来解决预测目标和真实目标之间的匹配问题。DETR 将真值分配问题转化为一个动态过程，但由于其离散的二分匹配和随机的训练过程，导致了一个匹配不稳定问题。有研究 [4] 表明由于阻塞对的存在，匈牙利匹配可能无法产生稳定的匹配结果。即使是代价矩阵的微小变动，都可能导致匹配结果的显著变化，从而使得解码器在不同周期中对查询的优化目标不一致。

本文将 DETR-like 模型的训练过程分为学习“好的锚点”和学习相对偏移两个阶段。解码器查询负责学习锚点，锚的更新不一致会使学习相对偏移量变得困难，因此，为了解决这个问题，本文提出了一种新的训练策略，即通过引入去噪任务来简化训练过程。这个去噪任务作为一个辅助任务，使得学习相对偏移量变得更加容易，因为它避免了二分图匹配的复杂性。由于我们将每个解码器查询解释为一个 4D 锚框，所以一个噪声查询可以被视为一个“好的锚点”，其附近有一个对应的真值框。因此，去噪训练的目标变得明确，即预测出原始的边界框，这从根本上消除了由匈牙利匹配引入的不确定性。

对于每个训练图像，收集所有真实目标 GT 对象，并在它们的边界框和类别标签上添加随机噪声。噪声分为两种方式：中心偏移和边界框缩放。中心偏移即在边界框的中心点添加随机噪声（Δx, Δy），确保偏移量不会使噪声化的边界框中心点超出原始边界框。具体来说，偏移量被限制在边界框宽度和高度的一定比例（$\lambda 1$）内。边界框缩放即随机缩放边界框的宽度和高度。宽度和高度分别在 [(1 - $\lambda 2$) * w, (1 + $\lambda 2$) * w] 和 [(1 - $\lambda 2$) * h, (1 + $\lambda 2$) * h] 的范围内随机采样，其中 $\lambda 2$ 是缩放的超参数。

3.5 损失函数定义

DN-DETR 模型使用了两种损失函数：一种是用于去噪任务的重建损失，另一种是用于匹配任务的匈牙利损失。

重建损失：对于去噪部分，模型需要从噪声化的 GT 对象中恢复出原始的 GT 对象。为了实现这一目标，模型需要最小化预测结果与真实 GT 对象之间的差异。这里使用 L1 损失和 GIoU 损失来重建边界框，使用 Focal 损失来重建类别标签。其中 L1 损失用于衡量预测的边界框坐标与真实边界框坐标之间的绝对差异。GIoU 损失用于衡量预测的边界框与真实边

界框之间的交并比（IoU）差异。Focal 损失用于类别预测，它通过引入一个调制因子来调整容易样本和困难样本的权重，使得模型更加关注于难以分类的样本。

匈牙利损失：对于匹配部分，模型使用匈牙利损失来优化预测的集合与真实集合之间的匹配。匈牙利损失通过最小化预测集合与真实集合之间不匹配的总代价来实现优化。

4 复现细节

4.1 与已有开源代码对比

本文任务基于原论文的开源代码进行复现。源代码中涉及到数据处理、模型训练和评估几个部分的代码，复现过程根据要求从官网上获取 COCO 数据集，然后选择预训练模型对数据集进行训练和评估。由于原论文对于 DN-DETR 模型的性能比较进行了多个任务的实验，考虑到时间和设备因素，根据复现计划选择部分任务进行复现，并对源代码进行微调，包括删除修改代码、参数设置等。

4.2 实验环境搭建

本实验采用的深度学习框架为 PyTorch，编程环境是 Python 3.7。机器配置 CPU 为 Intel(R) Core(TM) i5-8265U CPU@1.60GHz，内存 8GB，GPU 为 NVIDIA GeForce RTX 3090，显存 24GB，操作系统为 Windows 10。

前期准备工作：参考相关源代码，配置好相应的 cuda 和 pytorch 环境，cuda=11.1.1，pytorch=1.9.0，Torchvision=0.10.0，安装实验所需的库，编译 CUDA 运算符。下载 COCO 数据集，实验在具有挑战性的 MS-COCO 2017 目标检测任务上进行。MS-COCO 数据集包含 160K 张图像，分为训练集（train2017）、验证集（val2017）和测试集（test2017）。

4.3 模型训练

本文将基于 DAB-DETR 测试去噪训练方法的有效性，DAB-DETR 是由一个 CNN 主干、多个 Transformer 编码器层和解码器层组成的。本文采用在 ImageNet 上预训练的多个 ResNet 模型作为主干网络，并在 ResNet-50(R50) 上进行实验对比。对于超参数，遵循 DAB-DETR 使用 6 层 Transformer 编码器和 6 层 Transformer 解码器，并使用 256 作为隐藏层通道数。在 box 上添加均匀噪声，并设置关于噪声的超参数为 $\lambda 1 = 0.4$，$\lambda 2 = 0.4$，$\gamma = 0.2$。对于学习率调度器，使用初始学习率 lr=1×10^{-4} 并在第 40 个 epoch 后，将 lr 乘以 0.1 作为 50-epoch 的设置，在第 11 个 epoch，将 lr 乘以 0.1 作为 12-epoch 的设置。使用权重衰减为 1×10^{-4} 的 AdamW 优化器，batch size 大小为 16。

在对比实验中，将 DN-DETR 与 DAB-DETR 以及其他 DETR 的基线版本进行了测试结果比较。此外，基于 Deformable DETR 和 DAB 做 DN-Deformable-DETR 来在 COCO 上进行实验，比较模型的训练速度和优越性。为了最大程度验证去噪训练策略的提升，首先在标准的 50epoch 训练下进行完全相同 setting 并相比 DAB-DETR 在各个不同 backbone 上的提升，通过比较不同模型在相同设置下的性能，展示了去噪训练的有效性。实验评估指标使用标准的均值平均精度（AP）来评估模型性能，该指标在 COCO 验证集上根据不同的 IoU 阈值和对象尺度进行计算。

4.4 创新点

引入去噪训练：首次将去噪训练的概念引入到目标检测模型中，特别是在 DETR-like 模型的训练过程中。通过在训练过程中添加噪声并训练模型从噪声中恢复出原始的真实边界框，这种方法有效地稳定了二分图匹配过程，加速了模型的训练收敛。

通用性:提出的方法具有很高的通用性,可以通过简单的代码修改轻松集成到任何 DETR-like 模型中。这意味着去噪训练可以作为一种通用的训练策略，应用于多种不同的目标检测和分割模型。

性能提升：实验结果表明，去噪训练不仅加速了模型的训练过程，还显著提高了模型的性能。在相同的设置下，去噪训练使得模型在更少的训练周期内达到了与基线模型相当的性能，甚至在某些情况下超过了基线模型。

5 实验结果分析

5.1 模型在 ResNet-50 上的评估结果

使用标准 DN-DETR-R50 和 DN-Deformable-DETR-R50 预训练模型在 ResNet-50 主干网络上进行训练和评估，通过 AP 值来评估模型性能，得到的结果如图 3、 4所示:

```
DONE (t=34.33s).
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.442
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.642
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.472
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.230
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.483
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.636
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.357
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.578
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.628
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.388
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.690
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.855
```

图 3. DN-DETR-R50 的训练 AP 结果

```
Accumulating evaluation results...
DONE (t=14.10s).
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.494
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.675
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.540
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.313
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.526
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.656
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.375
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.629
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.674
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.476
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.719
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.858
```

图 4. DN-Deformable-DETR-R50 的训练 AP 结果

IoU 是目标检测中常用的评价指标，用于衡量预测边界框和真实边界框之间的重叠程度。IoU 值的范围从 0 到 1，值越高表示重叠程度越大。

Average Precision (AP) 是目标检测中常用的评价指标，它衡量模型在不同召回率（Recall）下的平均精度。AP 值越高，表示模型的性能越好。AP@IoU=0.50:0.95 是指在 IoU 阈值范围从 0.50 到 0.95 时计算的平均精度。这个范围覆盖了从松散匹配到严格匹配的不同情况，提供了模型性能的全面评估。AP@IoU=0.50 是在 IoU 阈值为 0.50 时计算的平均精度。这个阈值通常用于评估模型在较宽松匹配条件下的性能。而 AP@IoU=0.75 是在 IoU 阈值为 0.75 时计算的平均精度。这个阈值用于评估模型在较严格匹配条件下的性能。同时展示了 area 在 all、小、中、大尺寸目标上的 AP 值。这有助于理解模型在处理不同大小目标时的性能差异。

Average Recall (AR) 是另一个常用的目标检测评价指标，它衡量模型在不同精度（Precision）下的平均召回率。AR 值越高，表示模型在不同精度水平下召回目标的能力越强。通过比较在不同 IoU 阈值和最大检测数（maxDets）下的 AR 值可以帮助理解模型在不同检测约束下的性能。

从两个模型的结果来看，DN-DETR-R50 模型得到的 AP 值为 44.2，DN-Deformable-DETR-R50 模型得到的 AP 值为 49.4。后者在整体性能上略优于第前者。

5.2 模型在标准 50 epoch 设置下的性能比较

图 5显示原论文中 DN-DETR 和其他检测模型在相同设置下的结果。除了 DETR 之外，所有类似 DETR 的模型都使用 300 个查询，而 DETR 使用 100 个查询。

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

图 5. DN-DETR 和其他检测模型在相同设置下的结果

使用 DN-DETR-R50 和 DN-DETR-DC5-R50 模型在标准 50 epoch 设置下进行实验，得到的 AP 值结果如图 6所示：

Model	Epoch	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
DN-DETR-R50	50	44.2	64.3	46.9	22.7	48.2	63.5
DN-DETR-DC5-R50	50	46.2	66.5	49.7	26.4	49.8	64.4

图 6. DN-DETR-R50 和 DN-DETR-DC5-R50 模型在标准 50 epoch 设置下的实验结果

根据图 5、图 6 的结果，使用 DN-DETR-R50 和 DN-DETR-DC5-R50 模型在标准 50 epoch 设置下复现的实验结果与原论文中的结果相近，达到复现目标。从原文结果可以看出，与其他模型相比，DN-DETR 模型在相同的设置下，训练周期显著减少（例如，从 500 个周期减少到 25 个周期），同时保持或提高了模型的性能。这表明 DN-DETR 通过引入去噪训练，有效地加速了模型的训练过程，并在某些情况下提高了模型的性能。此外，DN-DETR 模型在不同大小的目标上都显示出了较好的性能，这表明去噪训练有助于模型更好地泛化到不同尺寸的目标。

5.3 不同模型的 Epoch-AP 收敛曲线图

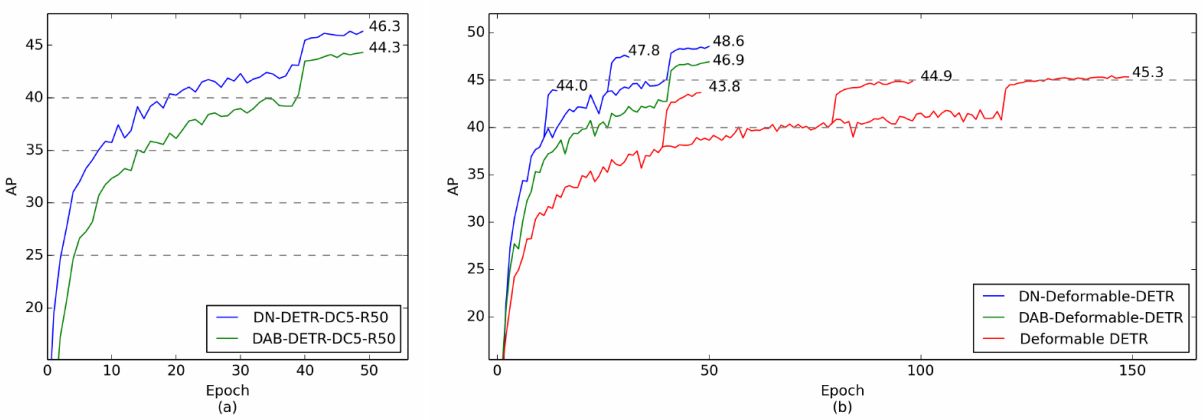

图 7. Epoch-AP 曲线图

对比使用 ResNet-DC5-50 主干网络下的 DAB-DETR 和 DN-DETR 的收敛曲线，以及 DN-Deformable-DETR、DAB-Deformable-DETR 和 Deformable-DETR 在使用 ResNet-50 的多尺度模型的收敛曲线。从图 7（a）可以看出，在学习速率下降之前，DN-DETR 在 20 个 epoch 内达到 40 个 AP，而 DAB-DETR 需要 40 个 epoch。这表明 DN-DETR 在相同的网络结构下，训练速度更快，收敛更快。从（b）可以看出，在训练过程中，DN-Deformable-DETR 在 30 个 epoch 时达到了 47.8 AP，比 DAB-Deformable-DETR 高 0.9 AP。这表明 DN-DETR 在多尺度设置下也显示出更快的收敛速度和更高的性能。

通过这些曲线，可以观察到 DN-DETR 相比于 DAB-DETR 和其他基线模型，在训练效率和最终性能上都有显著的提升。也体现了 DN-DETR 模型的优越性。

6 总结与展望

本文提出了一种新的去噪训练方法 DN-DETR，旨在加速 DETR（DEtection TRansformer）的训练过程。研究者发现 DETR 训练收敛慢的问题源于二分图匹配的不稳定性，这在训练的早期阶段尤为明显。为了解决这个问题，DN-DETR 在 DETR 的基础上引入了去噪任务，通过在 Transformer 解码器中添加带有噪声的真实边界框（GT bounding boxes），训练模型重建原始框。这种方法有效地降低了二分图匹配的难度，从而加快了收敛速度。DN-DETR 的方法是通用的，可以通过添加少量代码轻松集成到任何 DETR-like 方法中。实验结果表明，DN-DETR 在相同的设置下实现了显著的性能提升，并且在更少的训练周期内达到了与基线相当的性能。

这次的复现任务主要探究了 DN-DETR 模型的性能，以及其对比与其他基础模型的实验结果，检验去噪训练策略的有效性。由于多个模型和实验对比的训练时间超过了设备的支持，因此，选择具有代表性的对比实验进行复现和改进。

未来的工作可以探索更复杂的噪声模式，以进一步提高模型在复杂场景下的鲁棒性。研究如何将去噪训练策略扩展到其他类型的计算机视觉任务，如分割、姿态估计等。可以进一步优化去噪训练的参数设置，如噪声的强度和分布，以找到最佳的训练策略。还可以探索去噪训练在半监督学习和迁移学习中的应用，以提高模型在有限标注数据下的性能。

参考文献

[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] Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. *arXiv preprint arXiv:2109.10852*, 2021.

[3] 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.

[4] 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*, 917:1–79, 2021.

[5] 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.

[6] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3651–3660, 2021.

[7] 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.

[8] Zhiqing Sun, Shengcao Cao, Yiming Yang, and Kris M Kitani. Rethinking transformer-based set prediction for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3611–3620, 2021.

[9] 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.

[10] 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.