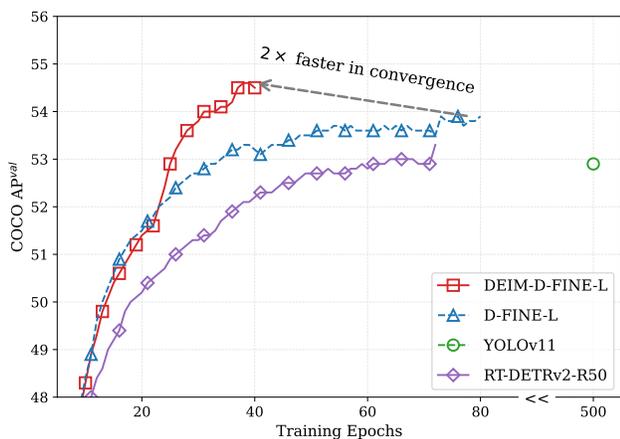


# DEIM: 改进匹配的 DETR 以实现快速收敛

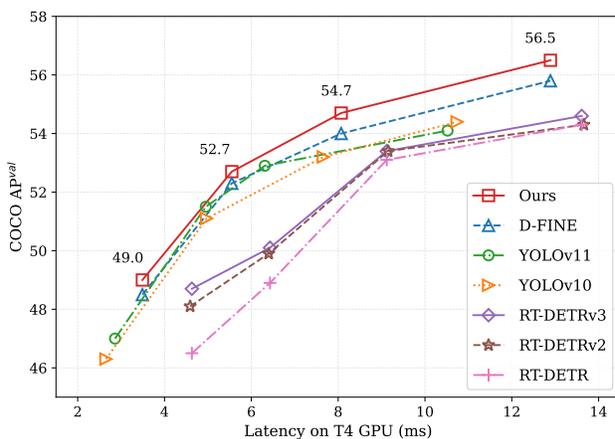
Shihua Huang<sup>1†</sup> Zhichao Lu<sup>2</sup> Xiaodong Cun<sup>3</sup> Yongjun Yu<sup>1</sup> Xiao Zhou<sup>4</sup> Xi Shen<sup>1✉</sup>

<sup>1</sup>Intellindust AI Lab <sup>2</sup>City University of Hong Kong <sup>3</sup>Great Bay University <sup>4</sup>Hefei Normal University

✉ Corresponding author: shenxiluc@gmail.com; † Project lead.



(a) 更快: 训练更加计算高效



(b) 更好: 超越所有实时检测器

Figure 1. Comparison with state-of-the-art real-time object detectors on COCO [19]. The proposed DEIM achieves faster convergence (a) and superior performance in terms of average precision (AP) and latency (b) when compared to state-of-the-art real-time object detectors.

## Abstract

我们介绍了 DEIM, 一个创新且高效的训练框架, 旨在加速基于 Transformer 架构 (DETR) 的实时目标检测的收敛速度。为了缓解 DETR 模型中固有的一对一 (O2O) 匹配的稀疏监督, DEIM 采用了密集 O2O 匹配策略。通过使用标准的数据增强技术, 该方法通过引入额外的目标来增加每张图像的正样本数量。尽管密集 O2O 匹配加速了收敛速度, 但它也引入了许多可能影响性能的低质量匹配。为了解决这个问题, 我们提出了可匹配性感知损失 (MAL), 这是一种新颖的损失函数, 可以优化各种质量水平的匹配, 从而提高密集 O2O 的有效性。在 COCO 数据集上进行的大量实验验证了 DEIM 的有效性。当与 RT-DETR 和 D-FINE 结合时, 它不仅持续提升性能, 还将训练时间减少了 50%。值得注意的是, 与 RT-DETRv2 配对时, DEIM 在 NVIDIA 4090 GPU 上一天的训练中达到 53.2% AP。此外, 使用 DEIM 训练的实时模型优于领先的实时目标检测器, DEIM-D-FINE-L 和 DEIM-D-FINE-X 在 NVIDIA T4 GPU 上分别以 124 和 78 FPS 达到 54.7% 和 56.5% AP, 无需额外数据。我们相信 DEIM 为实时目标检测

的进展设立了新的基准。我们的代码和预训练模型可在 <https://github.com/ShihuaHuang95/DEIM> 获得。

## 1. 介绍

目标检测是计算机视觉中的一项基础任务, 广泛应用于自动驾驶 [5, 6]、机器人导航 [9] 等领域。日益增长的高效检测器需求推动了实时检测方法的发展。特别是, YOLO 由于其在延迟和准确性之间的出色权衡 [1, 27, 31, 33, 43], 成为实时目标检测的主要范式之一。YOLO 模型被广泛认为是基于卷积神经网络的单阶段检测器。YOLO 系列中广泛采用了一对多 (O2M) 分配策略 [1, 27, 33, 43], 其中每个目标框与多个锚点相关联。这种策略被认为是有效的, 因为它提供了密集的监督信号, 加速了收敛并提升了性能 [43]。然而, 它为每个对象生成多个重叠的边界框, 需借助手工设计的非极大值抑制 (NMS) 来去除冗余, 从而引入延迟和不稳定性 [31, 42]。

基于 Transformer 的检测 (DETR) 范式的出现 [3] 引起了极大的关注 [4, 38, 45], 利用多头注意力捕捉全局上下文, 从而增强定位和分类。DETRs 采用一对

一 (O2O) 匹配策略, 在训练过程中利用匈牙利 [15] 算法在预测框和真实物体之间建立唯一对应关系, 消除了对 NMS 的需求。这种端到端框架为实时目标检测提供了一个引人注目的替代方案。

然而, 缓慢的收敛仍然是 DETRs 的主要限制之一, 我们假设原因有两个。1. 稀疏监督: O2O 匹配机制每个目标只分配一个正样本, 极大地限制了正样本的数量。相比之下, O2M 生成的正样本数量是其数倍。这种正样本的稀缺限制了密集监督, 从而阻碍了有效的模型学习——特别是对于小目标, 密集监督对性能至关重要。2. 低质量匹配: 与依赖密集 anchors (通常为  $> 8000$ ) 的传统方法不同, DETR 采用少量 (100 或 300 个) 随机初始化的查询。这些查询与目标在空间上缺乏对齐, 导致训练中出现大量低质量匹配, 即匹配的框与目标的 IoU 低但置信度高。

为了应对 DETR 中监督的稀缺, 近期的研究通过在 O2O 训练中引入 O2M 分配, 放宽了 O2O 匹配的约束, 从而为每个目标引入辅助的正样本以增加监督。Group DETR [4] 使用多个查询组来实现这一点, 每个组都有独立的 O2O 匹配, 而 Co-DETR [45] 则结合了来自物体检测器如 Faster R-CNN [28] 和 FCOS [30] 的 O2M 方法。虽然这些方法成功地增加了正样本的数量, 但它们也需要额外的解码器, 这增加了计算开销, 并有产生冗余高质量预测的风险, 正如传统检测器一样。相比之下, 我们提出了一种新颖而简洁的方法, 称为稠密的一对一 (Dense O2O) 匹配。我们的关键想法是增加每个训练图像中的目标数量, 从而在训练期间生成更多的正样本。值得注意的是, 这可以通过使用经典技术如 mosaic [1] 和 mixup [37] 增强来轻松实现, 这些技术在每张图像中生成额外的正样本, 同时保留一对一匹配框架。稠密的 O2O 匹配可以提供与 O2M 方法相媲美的监督水平, 而没有通常与 O2M 方法相关的复杂性和开销。尽管尝试通过先验 [17, 38, 42, 44] 改进查询初始化, 从而在物体周围生成更有效的查询分布。这些改进的初始化方法通常依赖于从编码器 [38, 42] 提取的有限特征信息, 倾向于将查询聚集在少数显著的物体周围。相比之下, 大多数非显著物体缺乏附近的查询, 导致低质量匹配。当使用稠密 O2O 时, 这个问题变得更加明显。随着目标数量的增加, 显著目标和非显著目标之间的差异增大, 尽管匹配数量总体增加, 但出现低质量匹配的情况也随之上升。在这种情况下, 如果损失函数在处理这些低质量匹配方面有限, 这种差异将持续存在, 阻碍模型实现更好的性能。

现有的用于 DETR 的损失函数, 如 Varifocal Loss (VFL), 是针对低质量匹配数量相对较少的密集锚点设计的。它们主要惩罚高质量匹配, 特别是具有高 IoU 但低置信度的匹配, 并且放弃低质量匹配。为了处理低质量匹配并进一步改进 Dense O2O, 我们提出了可匹配性感知损失 (MAL)。MAL 通过结合匹配查询和目标之间的 IoU 与分类置信度来调整惩罚。对于高质量匹配, MAL 的表现与 VFL 类似, 但它更加重视低质量匹配, 从而在训练过程中提高了有限正样本的利用率。此外, MAL 提供了一种比 VFL 更简单的数学公式。

所提出的 DEIM 结合了 Dense O2O 和 MAL 以创建一个有效的训练框架。我们在 COCO [19] 数据集上进行了广泛的实验, 以评估 DEIM 的有效性。图中的结果 1 (a) 显示, DEIM 显著加速了 RT-DETRv2 [23] 和 D-FINE [26] 的收敛, 同时也提高了性能。具体来说, 在训练轮数仅为一半的情况下, 我们的方法分别比 RT-DETRv2 和 D-FINE 提高了 0.2 和 0.6 AP。此外, 我们的方法使得在一块 4090 GPU 上训练一个基于 ResNet50 的 DETR 模型成为可能, 在一天内 (大约 24 个 epoch) 实现了 53.2 % mAP。通过引入更高效的模型, 我们还推出了一组新的实时检测器, 该检测器优于现有模型, 包括最新的 YOLOv11 [13], 为实时目标检测设定了新的最先进水平 (SoTA) (图 1 (b))。

本文的主要贡献总结如下:

- 我们介绍了 DEIM, 这是一种用于实时目标检测的简单且灵活的训练框架。
- DEIM 通过分别改进与 Dense O2O 和 MAL 的数量和质量来加速收敛。
- 通过我们的方法, 现有的实时 DETR 可以在减少一半训练成本的情况下获得更好的性能。具体来说, 我们的方法超过了 YOLOs, 并且在与 D-FINE 中的高效模型配对后, 建立了新的实时目标检测的 SoTA (最新技术水平)。

## 2. 相关工作

使用变压器 (DETR) 进行目标检测 [3] 代表了从传统的 CNN 架构到变压器的转变。通过使用匈牙利 [15] 损失进行一对一匹配, DETR 消除了手工制作的 NMS 作为后处理的需要, 并实现了端到端的目标检测。然而, 它存在收敛速度慢和计算密集的问题。

一对一匹配限制每个目标只有一个正样本, 提供的监督远少于一对多匹配, 阻碍了优化。一些研究探索了在一对一框架内增加监督的方法。例如, Group DETR 使用“组”的概念来近似一对多。它使用  $K$  组查询, 其中  $K > 1$ , 并在每个组内独立执行一对一匹配。这允许每个目标被分配  $K$  个正样本。然而, 为了防止组之间的通信, 每个组需要一个单独的解码层, 最终导致  $K$  个并行解码器。H-DETR 中的混合匹配方案类似于 Group DETR。Co-DETR 揭示了一对多分配方法有助于模型学习更具辨别性的特征信息, 因此提出了一种协同混合分配方案, 通过带有一对多标签分配的辅助头增强编码器表示, 就像 Faster R-CNN 和 FCOS 一样。现有方法旨在增加每个目标的正样本数量以增强监督。相比之下, 我们的 Dense O2O 探索了另一种方向——增加每个训练图像的目标数量以有效地增强监督。不像现有方法需要额外的解码器或头部, 从而增加训练资源消耗, 我们的方法是无计算成本的。

**优化低质量匹配** 稀疏和随机初始化的查询缺乏与目标的空间对齐, 导致大量低质量匹配, 阻碍了模型的收敛。几种方法已经将先验知识引入查询初始化中, 例如锚点查询 [34]、DAB-DETR [20]、DN-DETR [17] 和

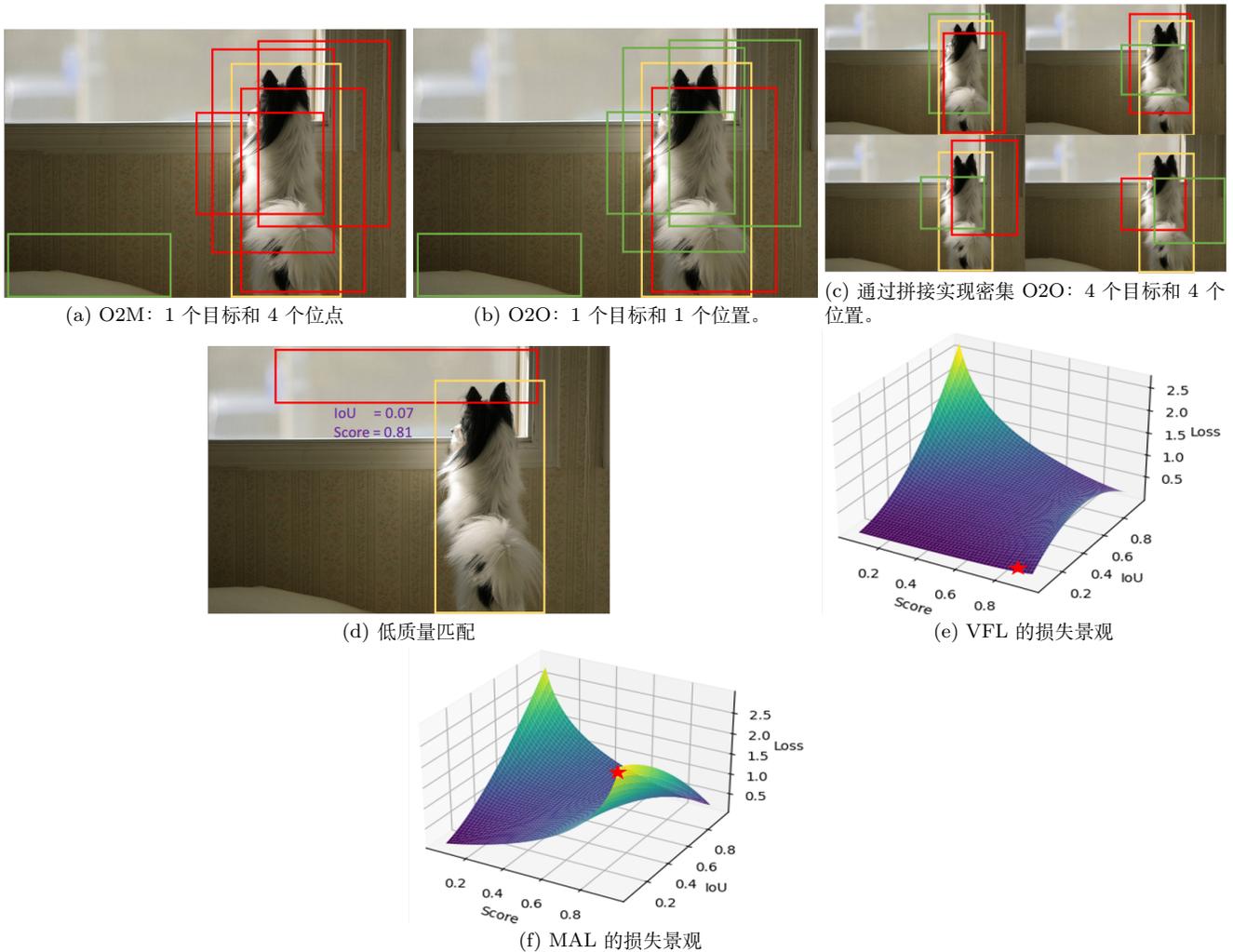


Figure 2. 我们提出的 DEIM 的示意图。黄色、红色和绿色分别代表 GT, 正样本和负样本。“pos.” 表示正样本。上图：我们密集 O2O(图 2c) 可以提供与 O2M(图 2a) 相同质量的正样本。下图：对于低质量匹配，使用 VFL [39] 和 MAL 的损失值由  $\star$  标记，表明 MAL 能够更有效地优化这些情况。

密集的独特查询 [40]。最近，受两阶段范式 [28, 44] 启发，诸如 DINO [38] 和 RT-DETR [42] 等方法利用编码器密集输出中的高排名预测来优化解码器查询 [35]。这些策略使得查询初始化更有效，且更接近目标区域。然而，低质量匹配仍然是一个显著的挑战 [21]。在 RT-DETR [42] 中，采用了 Varifocal Loss (VFL) 以减少分类置信度和框质量之间的不确定性，从而增强实时性能。然而，VFL 主要是为传统检测器设计的，这些检测器有较少的低质量匹配，并侧重于高 IoU 优化，低 IoU 匹配由于其最小和扁平的损失值而未得到充分优化。基于这些先进的初始化，我们引入了一种具有匹配感知能力的损失，以更好地优化不同质量级别的匹配，显著提升 Dense O2O 匹配的效果。

**减少计算成本。** 标准的注意力机制涉及密集计算。为了提高效率并促进与多尺度特征的交互，已经开发了

几种先进的注意力机制，如可变形注意力 [44]、多尺度可变形注意力 [41]、动态注意力 [7] 和级联窗口注意力 [36]。此外，最近的研究集中在创建更高效的编码器。例如，Lite DETR [16] 引入了一种在高层和底层特征之间交替更新的编码器块，而 RT-DETR [42] 在其编码器中结合了 CNN 和自注意力。两种设计显著减少了资源消耗，尤其是 RT-DETR。RT-DETR 是第一个在 DETR 框架内的实时目标检测模型。在此混合编码器的基础上，D-FINE [26] 通过额外的模块进一步优化 RT-DETR，并通过迭代更新概率分布而不是预测固定坐标来细化回归过程。这种方法使 D-FINE 得以在延迟和性能之间实现更佳的权衡，略微超越了最近的 YOLO 模型。利用这些在实时 DETR 中的进步，我们的方法在降低训练成本的同时实现了令人印象深刻的性能，在实时目标检测中大幅超越了 YOLO 模型。

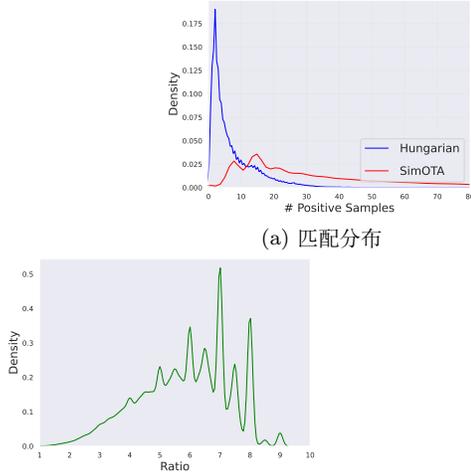


Figure 3. 锚点/查询匹配比较。使用一对多 (SimOTA [43]) 和一对一 (Hungarian [3]) 匹配方案在一个 COCO 周期中每张图片匹配的锚点/查询数量比较。

### 3. 方法

#### 3.1. 预备知识

**O2M 与 O2O 比较** O2M 分配策略 [10, 43] 被广泛应用于传统的目标检测器中，其监督可以表述如下：

$$\text{loss} = \sum_{i=0}^N \sum_{j=0}^{M_i} f(\hat{y}_{ij}, y_i), \quad (1)$$

，其中  $N$  是目标的总数， $M_i$  是第  $i$  个目标的匹配数量， $\hat{y}_{ij}$  表示第  $i$  个目标的第  $j$  个匹配， $y_i$  表示第  $i$  个真实标签， $f$  是损失函数。O2M 通过增加  $M_i$  来增强监督，即为每个目标分配多个查询 ( $M_i > 1$ )，从而提供密集的监督，如图 2a 所示。相比之下，O2O 分配仅将每个目标与单个最佳预测配对，通过匈牙利算法确定，该算法最小化平衡分类和定位误差的成本函数 (图 2b)。O2O 可以被认为所有目标的  $M_i = 1$  情况下的 O2M 的特例。

焦点损失 (FL) [18] 被引入是为了防止大量简单的负样本在训练期间压倒检测器，而是将关注点引导到一小部分困难样本上。它作为基于 DETR 的检测器中的默认分类损失 [38, 44]，并定义如下：

$$\text{FL}(p, y) = \begin{cases} -\alpha(1-p)^\gamma \log(p) & y = 1 \\ -(1-\alpha)p^\gamma \log(1-p) & y = 0, \end{cases} \quad (2)$$

，其中  $y \in \{0, 1\}$  指定了真实类别， $p \in [0, 1]$  表示前景类别的预测概率。参数  $\gamma$  控制简单和困难样本之间的平衡，而  $\alpha$  调整前景和背景类别之间的权重。在 FL 中，仅考虑样本的类别和置信度，不关注边界框质量，即定位。

#### 3.2. 提高匹配效率：密集 O2O

在 DETR 模型中普遍使用的一对一 (O2O) 匹配方案，将每个目标只匹配到一个预测查询。这个方法通过匈牙利算法 [15] 实现，使得端到端训练成为可能，并且消除了对 NMS 的需求。然而，O2O 的一个关键限制是，相比传统的一对多 (O2M) 方法如 SimOTA [43]，它生成的正样本显著减少。这导致了监督的稀疏性，从而可能在训练过程中减慢收敛速度。

为了更好地理解这个问题，我们在 MS COCO 数据集上使用 ResNet50 骨干网络训练了 RT-DETRv2 [23]。我们比较了匈牙利 (O2O) 和 SimOTA (O2M) 策略产生的正样本匹配数量。如图 3a 所示，O2O 在每张图像下产生的正样本匹配数量在 10 个以下出现了一个峰值，而 O2M 产生了更宽分布，有许多更多的正样本匹配，有时单张图像超过 80 个正样本。图 3b 进一步指出，在极端情况下，SimOTA 生成的匹配数量约为 O2O 的 10 倍。这表明 O2O 的正样本匹配较少，可能会减缓优化过程。

我们提出了 Dense O2O 作为一种高效的替代方案。这一策略保留了 O2O (一对一) 匹配结构 (具有  $M_i = 1$ )，但增加了每张图片的目标数量 ( $N$ )，实现了更密集的监督。例如，如图 2c 所示，我们将原始图像复制到四个象限，并将它们合并成一个复合图像，保持原始图像的尺寸。这样将目标数量从 1 增加到 4，在方程 1 中提升了监督水平，同时保持匹配结构不变。Dense O2O 在监督水平上达到了与 O2M 相当的效果，但没有额外的复杂性和计算开销。

#### 3.3. 改进匹配质量：匹配性感知损失

**VFL 的局限性。** 变焦损失 (VFL) [39]，基于 FL [18]，已被证明可以提高目标检测性能，尤其是在 DETR 模型中 [2, 23, 42]。VFL 损失表示为：

$$\text{VFL}(p, q, y) = \begin{cases} -q(q \log(p) + (1-q) \log(1-p)) & q > 0 \\ -\alpha p^\gamma \log(1-p) & q = 0, \end{cases} \quad (3)$$

其中  $q$  表示预测边界框与其目标框之间的 IoU。对于前景样本 ( $q > 0$ )，目标标签设置为  $q$ ，而背景样本 ( $q = 0$ ) 的目标标签为 0。VFL 结合了 IoU，以提高 DETR 中查询的质量 [42]。

然而，VFL 在优化低质量匹配时有两个主要限制：i) 低质量匹配。VFL 主要关注高质量匹配 (高 IoU)。对于低质量匹配 (低 IoU)，损失保持较小，阻止模型对低质量框进行细化预测。然而，对于低质量匹配 (低 IoU，例如如图 2d 所示)，损失仍然最小 (在图 2e 中用  $\star$  表示)。ii) 负样本。VFL 将没有重叠的匹配视为负样本，这减少了正样本的数量并限制了有效训练。

由于传统检测器具有密集的锚点和多种分配策略，这些问题较不严重。然而，在 DETR 框架中，由于查询是稀疏的且匹配更为严格，这些限制变得更加显著。

为了解决这些问题，我们提出了可匹配性感知损失 (MAL)，它在扩展 VFL 优点的同时减轻了其缺点。

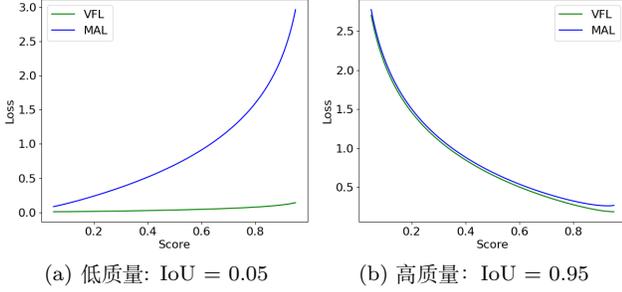


Figure 4. VFL 与 MAL 对比。在低质量 (IoU = 0.05, 图 4a) 和高质量 (IoU = 0.95, 图 4b) 匹配情况下, 比较 VFL 和我们的方法 MAL。

MAL 将匹配质量直接纳入损失函数, 使其对低质量匹配更加敏感。的公式为:

$$\text{MAL}(p, q, y) = \begin{cases} -q^\gamma \log(p) + (1 - q^\gamma) \log(1 - p) & y = 1 \\ -p^\gamma \log(1 - p) & y = 0. \end{cases} \quad (4)$$

与 VFL 相比, 我们引入了几个小但重要的变化。具体来说, 目标标签已从  $q$  修改为  $q^\gamma$ , 简化了正负样本的损失权重并移除了用于平衡正负样本的超参数  $\alpha$ 。此更改有助于避免过度强调高质量边框并改进整体训练过程。从 VFL (在图 2e 中) 和 MAL (在图 2f 中) 之间的损失曲面可以轻易看出这一点。注意,  $\gamma$  的影响在第 ?? 节中提供。

我们比较了 MAL 和 VFL 在处理低质量和高质量匹配时的表现。在低质量匹配 (IoU = 0.05, 见图 4a) 的情况下, 与几乎没有变化的 VFL 相比, MAL 在预测置信度增加时的损失上升更为明显。对于高质量匹配 (IoU = 0.95, 见图 4b), MAL 和 VFL 的表现类似, 这证实了 MAL 在不影响高质量匹配性能的情况下提高了训练效率。

## 4. 实验

对于 Dense O2O, 我们应用了 mosaic 增强 [1] 和 mixup 增强 [37] 来为每张图像生成额外的正样本。在第 ?? 节中讨论了这些增强的影响。我们在 MS-COCO 数据集 [19] 上使用 AdamW 优化器 [22] 训练我们的模型。使用了标准的数据增强方法, 如颜色抖动和缩放, 就像在 RT-DETR [23, 42] 和 D-FINE [26] 中一样。我们采用了一个平坦的余弦学习率调度器 [24] 并提出了一种新颖的数据增强调度器。在前几个训练周期 (通常是四个) 中使用数据增强预热策略, 以简化注意力学习。在训练周期的 50% 后禁用 Dense O2O 能够带来更好的结果。按照 RT-DETRv2 [42] 的方法, 我们在最后两个周期关闭数据增强。我们的 LR 和 DataAug 调度器具体描述在图 5 中。我们的主干网络在 ImageNet1k [8] 上进行了预训练。我们在分辨率为  $640 \times 640$  的 MS-COCO 验证集上评估我们的模型。关于超参数的更多细节, 提供在补充材料中。

我们将我们的方法整合到 D-FINE-L [26] 和 D-FINE-X [26] 中, 构建我们的 DEIM-D-FINE-L 和 DEIM-D-FINE-X。然后我们评估这些模型并将它们的实时目标检测性能与最新的模型进行基准测试, 包括 YOLOv8 [12], YOLOv9 [33], YOLOv10 [33], YOLOv11 [13], 以及基于 DETR 的模型如 RT-DETRv2 [23] 和 D-FINE [26]。表 1 对模型在训练周期、参数、GFLOPs、延迟和检测准确率方面进行了比较。补充材料中包含了较小模型变体 (S 和 M) 的额外比较。

我们的方法在训练成本、推理延迟和检测准确性方面优于当前最先进的模型, 为实时目标检测设立了新的基准。需要注意的是, D-FINE [26] 是一个非常新的工作, 它通过结合蒸馏和边界框细化来增强 RT-DETRv2 [23] 的性能, 确立了自己作为领先的实时检测器的地位。我们的 DEIM 进一步提升了 D-FINE 的性能, 实现了 0.7 AP 的提升, 同时将训练成本降低了 30%, 且没有增加推理延迟。最显著的改进体现在小目标检测上, 在使用我们的方法进行训练时, D-FINE-X [26] 相较于 DEIM-D-FINE-X 实现了 1.5 AP 的提升。

与 YOLOv11-X [13] 直接对比时, 我们的方法优于这一最先进的 YOLO 模型, 取得了稍高的性能 (54.7 对 54.1 的 AP) 并将推理时间减少了 20% (8.07 毫秒对 10.74 毫秒)。虽然 YOLOv10 [33] 使用了混合的 O2M 和 O2O 分配策略, 我们的模型仍然始终优于 YOLOv10, 这证明了我们 Dense O2O 策略的有效性。

尽管在小物体检测方面相对于其他 DETR 模型有显著改进, 但我们的方法在小物体 AP 方面相比 YOLO 模型略有下降。例如, YOLOv9-E [33] 在小物体上比 D-FINE-L [26] 高约 1.4 AP, 尽管我们的模型实现了更高的整体 AP (56.5 vs. 55.6)。这一差距突显了 DETR 架构中小物体检测的持续挑战, 并表明了进一步改进的潜在方向。

大多数 DETR 研究使用 ResNet [14] 作为主干网络, 为了在现有 DETR 变体之间进行全面比较, 我们的方法也应用于最新的 DETR 变体 RT-DETRv2 [23]。结果总结如表 2 所示。与原始 DETR 需要 500 个周期来进行有效训练不同, 包括我们的在内的最近 DETR 变体在减少训练时间的同时提高了模型性能。我们的方法显示出最显著的改进, 仅在 36 个周期后就超过了所有变体。具体来说, 将训练时间减少了一半, 且在 RT-DETRv2 [23] 上分别使用 ResNet-50 [14] 和 ResNet-101 [14] 主干网络时, AP 提高了 0.5 和 0.9。此外, 与 ResNet-50 [14] 主干网络的 DINO-Deformable-DETR [38] 相比, AP 提高了 2.7。

DEIM 也显著增强了小物体检测。例如, 在实现与 RT-DETRv2 [23] 相当的整体 AP 时, 我们的 DEIM-RT-DETRv2-R50 在小物体上比 RT-DETRv2 超出 1.3 AP。使用更大的 ResNet-101 主干网时, 这种改进更加明显, 我们的 DEIM-RT-DETRv2-R101 在小物体上比 RT-DETRv2-R101 超过 2.1 AP。将训练拓展到 72 轮进一步提升了整体性能, 特别是对于 ResNet-50

Table 1. 在 COCO [19] val2017 上与实时目标检测器的比较。通过将我们的方法集成到 D-FINE-L [26] 和 D-FINE-X [26] 中，我们构建了 DEIM-D-FINE-L 和 DEIM-D-FINE-X。我们将我们的方法与基于 YOLO 和基于 DETR 的实时目标检测器进行比较。\* 表示 NMS 以 0.01 的置信度阈值进行调整。

Model	# Epochs	# Params	GFLOPs	Latency (ms)	AP <sup>val</sup>	AP <sup>val</sup> <sub>50</sub>	AP <sup>val</sup> <sub>75</sub>	AP <sup>val</sup> <sub>S</sub>	AP <sup>val</sup> <sub>M</sub>	AP <sup>val</sup> <sub>L</sub>
YOLO-based Real-time Object Detectors										
YOLOv8-L [12]	500	43	165	12.31	52.9	69.8	57.5	35.3	58.3	69.8
YOLOv8-X [12]	500	68	257	16.59	53.9	71.0	58.7	35.7	59.3	70.7
YOLOv9-C [33]	500	25	102	10.66	53.0	70.2	57.8	36.2	58.5	69.3
YOLOv9-E [33]	500	57	189	20.53	55.6	72.8	60.6	40.2	61.0	71.4
Gold-YOLO-L [32]	300	75	152	9.21	53.3	70.9	-	33.8	58.9	69.9
YOLOv10-L * [31]	500	24	120	7.66	53.2	70.1	58.1	35.8	58.5	69.4
YOLOv10-X * [31]	500	30	160	10.74	54.4	71.3	59.3	37.0	59.8	70.9
YOLO11-L * [13]	500	25	87	6.31	52.9	69.4	57.7	35.2	58.7	68.8
YOLO11-X * [13]	500	57	195	10.52	54.1	70.8	58.9	37.0	59.2	69.7
DETR-based Real-time Object Detectors										
RT-DETR-HG-L [42]	72	32	107	8.77	53.0	71.7	57.3	34.6	57.4	71.2
RT-DETR-HG-X [42]	72	67	234	13.51	54.8	73.1	59.4	35.7	59.6	72.9
D-FINE-L [26]	72	31	91	8.07	54.0	71.6	58.4	36.5	58.0	71.9
DEIM-D-FINE-L	50	31	91	8.07	54.7	72.4	59.4	36.9	59.6	71.8
D-FINE-X [26]	72	62	202	12.89	55.8	73.7	60.2	37.3	60.5	73.4
DEIM-D-FINE-X	50	62	202	12.89	56.5	74.0	61.5	38.8	61.4	74.2

Table 2. 与基于 ResNet 的 DETRs 在 COCO [19] val2017 上的比较。通过将我们的方法与 ResNet50 [14] 和 ResNet101 [14] 相结合，我们构建了 DEIM-RT-DETRv2-R50 和 DEIM-RT-DETRv2-R101。我们将我们的方法与使用 ResNet50 [14] 或 ResNet101 [14] 作为骨干网的有竞争力的基于 DETR 的目标检测器进行比较。

Model	# Epochs	# Params	GFLOPs	AP <sup>val</sup>	AP <sup>val</sup> <sub>50</sub>	AP <sup>val</sup> <sub>75</sub>	AP <sup>val</sup> <sub>S</sub>	AP <sup>val</sup> <sub>M</sub>	AP <sup>val</sup> <sub>L</sub>
ResNet50 [14]-based									
DETR-DC5 [3]	500	41	187	43.3	63.1	45.9	22.5	47.3	61.1
Anchor-DETR-DC5 [34]	50	39	172	44.2	64.7	47.5	24.7	48.2	60.6
Conditional-DETR-DC5 [25]	108	44	195	45.1	65.4	48.5	25.3	49.0	62.2
Efficient-DETR [35]	36	35	210	45.1	63.1	49.1	28.3	48.4	59.0
SMCA-DETR [11]	108	40	152	45.6	65.5	49.1	25.9	49.3	62.6
Deformable-DETR [44]	50	40	173	46.2	65.2	50.0	28.8	49.2	61.7
DAB-Deformable-DETR [20]	50	48	195	46.9	66.0	50.8	30.1	50.4	62.5
DAB-Deformable-DETR++ [20]	50	47	-	48.7	67.2	53.0	31.4	51.6	63.9
DN-Deformable-DETR [17]	50	48	195	48.6	67.4	52.7	31.0	52.0	63.7
DN-Deformable-DETR++ [17]	50	47	-	49.5	67.6	53.8	31.3	52.6	65.4
DINO-Deformable-DETR [38]	36	47	279	50.9	69.0	55.3	34.6	54.1	64.6
RT-DETR [42]	72	42	136	53.1	71.3	57.7	34.8	58.0	70.0
RT-DETRv2 [23]	72	42	136	53.4	71.6	57.4	36.1	57.9	70.8
DEIM-RT-DETRv2	36	42	136	53.9	71.7	58.6	36.7	58.9	70.9
DEIM-RT-DETRv2	60	42	136	54.3	72.3	58.8	37.5	58.7	70.8
ResNet101 [14]-based									
DETR-DC5 [3]	500	60	253	44.9	64.7	47.7	23.7	49.5	62.3
Anchor-DETR-DC5 [34]	50	-	-	45.1	65.7	48.8	25.8	49.4	61.6
Conditional-DETR-DC5 [25]	108	63	262	45.9	66.8	49.5	27.2	50.3	63.3
Efficient-DETR [35]	36	54	289	45.7	64.1	49.5	28.2	49.1	60.2
SMCA-DETR [11]	108	58	218	46.3	66.6	50.2	27.2	50.5	63.2
RT-DETR [42]	72	76	259	54.3	72.7	58.6	36.0	58.8	72.1
RT-DETRv2 [23]	72	76	259	54.3	72.8	58.8	35.8	58.8	72.1
DEIM-RT-DETRv2	36	76	259	55.2	73.3	59.9	37.8	59.6	72.8
DEIM-RT-DETRv2	60	76	259	55.5	73.5	60.3	37.9	59.9	73.0

主干网，这表明较小的模型从额外的训练中受益。

#### 4.1. 在 CrowdHuman 上的比较

CrowdHuman [29] 是一个旨在评估密集人群场景中物体检测器的基准数据集。我们按照官方库中的配置将

D-FINE 和我们提出的方法应用于 CrowdHuman 数据集。如表所示 3，我们的方法（增强的 D-FINE-L 与 DEIM）比 D-FINE-L 显著提高了 1.5 AP。特别是，我们的方法在小物体（AP<sub>s</sub>）和高质量检测（AP<sub>75</sub>）上提供了显著的性能提升（超过 3% 的改进），表明其在

Table 3. 比较了 D-FINE 和我们在 CrowdHuman 上的 DEIM [29]。两者都经过 120 个纪元的训练。

Method	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP <sub>s</sub>	AP <sub>m</sub>	AP <sub>l</sub>
D-FINE-L	56.0	87.2	59.4	29.0	46.1	54.6
w/ DEIM	57.5	87.6	62.9	33.2	48.7	55.7

Table 4. 不同马赛克和混合增强策略组合的密集 O2O 方法比较。概率值表示在训练期间每个小批量中应用马赛克和混合的可能性。

Mosaic Prob.	Mixup Prob.	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP <sub>s</sub>	AP <sub>m</sub>	AP <sub>l</sub>
Training 12 Epochs							
0.0	0.0	49.6	67.1	53.6	31.3	54.2	67.8
0.5	0.0	50.4	68.4	54.5	32.7	54.6	68.1
0.0	0.5	50.1	67.7	54.0	31.1	54.5	68.7
0.5	0.5	50.4	68.1	54.2	32.7	54.7	68.2
Training 24 Epochs							
0.0	0.0	51.7	69.5	55.8	32.8	56.4	69.7
0.5	0.0	51.9	70.1	55.9	34.9	56.1	69.3
0.0	0.5	51.5	69.4	55.5	33.2	56.3	69.3
0.5	0.5	52.5	70.6	56.7	34.9	57.1	70.1

Table 5.  $\gamma$  在 MAL 中的影响 (方程 4)。我们报告了在 COCO [19] val2017 上进行 24 个周期的性能。

$\gamma$	1.3	1.5	1.8	2.0
AP	52.2	52.4	52.1	51.9

挑战性场景中更准确地检测物体的能力。此外，这次实验强调了我们在不同数据集上的强大泛化能力，进一步确认了其鲁棒性。

在以下研究中，我们使用 RT-DETRv2 与 ResNet50 配对进行实验，并报告在 MS-COCO val2017 上的性能，除非另有说明，否则这是默认设置。

我们探索了两种实现 Dense O2O 的方法：mosaic [1] 和 mixup [37]。Mosaic 是一种数据增强技术，它将四张图像合并为一张，而 mixup 则以随机比例叠加两张图像。这两种方法都有效地增加了每张图像的目标数量，从而在训练过程中增强监督效果。

如表 4 所示，与没有目标增强的训练相比，在进行 12 个周期后，拼图和混合增量都带来了显著的提升，这突出了 Dense O2O 的有效性。此外，结合拼图和混合增量加速了模型的收敛，进一步强调了增强监督的好处。我们进一步跟踪了一个训练周期内每张图像中的正样本数量，结果如图 6 所示。与传统的 O2O 匹配相比，Dense O2O 显著增加了正样本的数量。

总体而言，Dense O2O 通过增加每张图像的目标数量来加强监督，从而加速模型的收敛。Mosaic 和 mixup 是实现这一目标的简单且计算效率高的技术，它们的有效性表明在训练过程中探索其他方法来增加目标数量存在进一步潜力。

表 5 的结果显示了不同  $\gamma$  值在经过 24 个周期后的 MAL 的效果。基于这些实验，我们凭经验将  $\gamma$  设置为 1.5，因为它表现最佳。

表 6 展示了两个核心组件的有效性：Dense O2O 和 MAL。Dense O2O 显著加速了模型的收敛，仅在 36 个

Table 6. 密集 O2O 和 MAL 的影响。我们使用 RT-DETRv2-R50 [23] 和 D-FINE-L [26] 进行实验。

Epochs	Dense O2O	MAL	AP	AP <sub>50</sub>	AP <sub>75</sub>
RT-DETRv2-R50 [23]					
72			53.4	71.6	57.4
36	✓		53.6	71.9	58.2
	✓	✓	53.9	71.7	58.6
D-FINE-L [26]					
72			54.0	71.6	58.4
36	✓		54.2	72.1	58.9
	✓	✓	54.6	72.2	59.5

epoch 后就达到了与基线相似的性能，而原始模型需要 72 个 epoch。与 MAL 结合时，我们的方法进一步提高了性能。这一改进主要得益于更好的框质量，与我们优化低质量匹配以改进高质量框预测的目标一致。总体而言，Dense O2O 和 MAL 在 RT-DETRv2 和 D-FINE 中始终如一地提升了性能，显示了它们的稳健性与通用性。

在本文中，我们介绍了 DEIM，这是一种通过改进匹配来加速基于 DEER 的实时目标检测器收敛的方法。DEIM 集成了 Dense O2O 匹配，这增加了每张图像的正样本数量，并结合了 MAL，一种旨在优化不同质量匹配并特别增强低质量匹配的新损失函数。该组合显著提高了训练效率，使 DEIM 能够在比如 YOLOv11 等模型更少的次数中实现卓越性能。DEIM 与 SoTA DETR 模型如 RT-DETR 和 D-FINE 相比，展现了明显的优势，在不影响推理延迟的情况下，显著提高了检测准确性和训练速度。这些特性使得 DEIM 成为一个高效的实时应用解决方案，并有可能进一步优化和应用到其他高性能检测任务。

感谢。感谢来自 Intellindust 的于宣龙、刘龙飞和谢海洋的友好讨论和有益建议。

## References

- [1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. arXiv, 2020.
- [2] Zhi Cai, Songtao Liu, Guodong Wang, Zheng Ge, Xiangyu Zhang, and Di Huang. Align-detr: Improving detr with simple iou-aware bce loss. In BMVC, 2024.
- [3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In ECCV, 2020.
- [4] Qiang Chen, Xiaokang Chen, Jian Wang, Shan Zhang, Kun Yao, Haocheng Feng, Junyu Han, Errui Ding, Gang Zeng, and Jingdong Wang. Group detr: Fast detr training with group-wise one-to-many assignment. In ICCV, 2023.
- [5] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In CVPR, 2016.
- [6] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In CVPR, 2017.
- [7] Xiyang Dai, Yinpeng Chen, Jianwei Yang, Pengchuan Zhang, Lu Yuan, and Lei Zhang. Dynamic detr: End-to-end object detection with dynamic attention. In ICCV, 2021.
- [8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In CVPR, 2009.
- [9] Andreas Ess, Konrad Schindler, Bastian Leibe, and Luc Van Gool. Object detection and tracking for autonomous navigation in dynamic environments. *The International Journal of Robotics Research*, 2010.
- [10] Chengjian Feng, Yujie Zhong, Yu Gao, Matthew R Scott, and Weilin Huang. Tood: Task-aligned one-stage object detection. In ICCV, 2021.
- [11] Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fast convergence of detr with spatially modulated co-attention. In ICCV, 2021.
- [12] Jocher Glenn. Yolov8. <https://docs.ultralytics.com/models/yolov8/>, 2023.
- [13] Jocher Glenn. Yolo11. <https://docs.ultralytics.com/models/yolo11/>, 2024.
- [14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016.
- [15] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 1955.
- [16] Feng Li, Ailing Zeng, Shilong Liu, Hao Zhang, Hongyang Li, Lei Zhang, and Lionel M Ni. Lite detr: An interleaved multi-scale encoder for efficient detr. In CVPR, 2023.
- [17] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In CVPR, 2022.
- [18] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In ICCV, 2017.
- [19] 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 ECCV, 2014.
- [20] 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 ICLR, 2022.
- [21] Shilong Liu, Tianhe Ren, Jiayu Chen, Zhaoyang Zeng, Hao Zhang, Feng Li, Hongyang Li, Jun Huang, Hang Su, Jun Zhu, et al. Detection transformer with stable matching. In ICCV, 2023.
- [22] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In ICLR, 2017.
- [23] Wenyu Lv, Yian Zhao, Qinyao Chang, Kui Huang, Guanzhong Wang, and Yi Liu. Rt-detrv2: Improved baseline with bag-of-freebies for real-time detection transformer. arXiv, 2024.
- [24] Chengqi Lyu, Wenwei Zhang, Haiyan Huang, Yue Zhou, Yudong Wang, Yanyi Liu, Shilong Zhang, and Kai Chen. RtmDET: An empirical study of designing real-time object detectors. arXiv, 2022.
- [25] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In ICCV, 2021.
- [26] Yansong Peng, Hebei Li, Peixi Wu, Yueyi Zhang, Xiaoyan Sun, and Feng Wu. D-fine: Redefine regression task in detr as fine-grained distribution refinement. arXiv, 2024.
- [27] J Redmon. You only look once: Unified, real-time object detection. In CVPR, 2016.
- [28] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. TPAMI, 2016.
- [29] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting human in a crowd. arXiv, 2018.
- [30] Zhi Tian, Xiangxiang Chu, Xiaoming Wang, Xiaolin Wei, and Chunhua Shen. Fully convolutional one-stage 3d object detection on lidar range images. In NIPS, 2022.
- [31] Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding. Yolov10: Real-time end-to-end object detection. 2024.
- [32] Chengcheng Wang, Wei He, Ying Nie, Jianyuan Guo, Chuanjian Liu, Yunhe Wang, and Kai Han. Gold-yolo: Efficient object detector via gather-and-distribute mechanism. NeurIPS, 2023.
- [33] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. Yolov9: Learning what you want to learn using programmable gradient information. arXiv, 2024.
- [34] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. In AAAI, 2022.
- [35] Zhuyu Yao, Jiangbo Ai, Boxun Li, and Chi Zhang. Efficient detr: improving end-to-end object detector

- with dense prior. arXiv, 2021.
- [36] Mingqiao Ye, Lei Ke, Siyuan Li, Yu-Wing Tai, Chi-Keung Tang, Martin Danelljan, and Fisher Yu. Cascade-detr: delving into high-quality universal object detection. In ICCV, 2023.
  - [37] Hongyi Zhang. mixup: Beyond empirical risk minimization. In ICLR, 2017.
  - [38] 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. In ICLR, 2023.
  - [39] Haoyang Zhang, Ying Wang, Feras Dayoub, and Niko Sunderhauf. Varifocalnet: An iou-aware dense object detector. In CVPR, 2021.
  - [40] Shilong Zhang, Xinjiang Wang, Jiaqi Wang, Jiangmiao Pang, Chengqi Lyu, Wenwei Zhang, Ping Luo, and Kai Chen. Dense distinct query for end-to-end object detection. In CVPR, 2023.
  - [41] Chuyang Zhao, Yifan Sun, Wenhao Wang, Qiang Chen, Errui Ding, Yi Yang, and Jingdong Wang. Ms-detr: Efficient detr training with mixed supervision. In CVPR, 2024.
  - [42] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. Dets beat yolos on real-time object detection. In CVPR, 2024.
  - [43] Ge Zheng, Liu Songtao, Wang Feng, Li Zeming, and Sun Jian. Yolox: Exceeding yolo series in 2021. arXiv, 2021.
  - [44] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In ICLR, 2021.
  - [45] Zhuofan Zong, Guanglu Song, and Yu Liu. Dets with collaborative hybrid assignments training. In ICCV, 2023.

# DEIM: 改进匹配的 DETR 以实现快速收敛

## Supplementary Material

### 5. 实验设置

数据集和指标。我们在 COCO [19] 数据集上评估我们的方法，在 train2017 上进行 DEIM 训练，并在 val2017 上进行验证。报告了标准的 COCO 指标，包括 AP (在 IoU 阈值从 0.50 到 0.95 之间，步长为 0.05 平均值)，AP<sub>50</sub>，AP<sub>75</sub>，以及在不同物体尺度上的 AP: AP<sub>S</sub>，AP<sub>M</sub> 和 AP<sub>L</sub>。

Table 7. 使用 DEIM 训练的 D-FINE 模型的不同超参数。

D-FINE	X	L	M	S
Base LR	5e-4	5e-4	4e-4	4e-4
Min LR	2.5e-4	2.5e-4	2e-4	2e-4
Backbone LR	5e-6	2.5e-5	4e-5	2e-4
Backbone MinLR	2.5e-6	1.25e-5	2e-5	1e-4
Weight of MAL	1	1	1	1
$\gamma$ in MAL	1.5	1.5	1.5	1.5
Freeze Backbone BN	False	False	False	False
Decoder Act.	SiLU	SiLU	SiLU	SiLU
Epochs	50	50	90	120

Table 8. 使用 DEIM 训练的 RT-DETRv2 模型的不同超参数。

RT-DETRv2	X	L	M *	M	S
Base LR	2e-4	2e-4	2e-4	2e-4	2e-4
Min LR	1e-4	1e-4	1e-4	1e-4	1e-4
Backbone LR	2e-6	2e-5	2e-5	1e-4	2e-4
Backbone MinLR	1e-6	1e-5	1e-5	5e-5	1e-4
Weight of MAL	1	1	1	1	1
$\gamma$ in MAL	1.5	1.5	1.5	1.5	1.5
Freeze Backbone BN	False	False	False	False	False
Decoder Act.	SiLU	SiLU	SiLU	SiLU	SiLU
Epochs	60	60	60	120	120

实施细节。我们使用 D-FINE [26] 和 RT-DETRv2 [23, 42] 框架实现和验证我们的方法。大多数超参数遵循它们的原始设置，不同的细节分别在表 7 和表 8 中详细说明。受到 RTMDet [24] 中 FlatCosine LR 调度器的启发，我们提出了一种专门为稠密 O2O 设计的新数据增强调度器。DETR 的注意机制对于提取准确的对象特征以进行定位和分类至关重要。然而，没有归纳偏差从头开始学习注意机制可能具有挑战性。为了解决这个问题，我们引入了一种数据增强热身策略，即 DataAug Warmup，通过在初始轮次禁用高级数据增强来简化学习。关于 60 个训练轮次的 FlatCosine LR 和提出的 DataAug 调度器的示例如图 5 所示。

我们在 Table 9 中展示了与更轻量级的实时模型 (S 和 M 大小) 的比较结果。基于强大的实时检测器

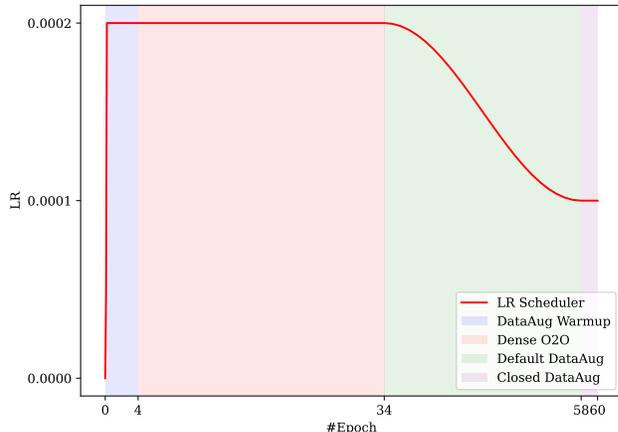


Figure 5. 我们提出的新颖训练方案的一个关于学习率和数据增强调度器的示例图解。

RT-DETRv2 [23] 和 D-FINE [26]，我们的 DEIM 在各方面都取得了显著的改进。值得注意的是，在 RT-DETRv2 中，所有三种模型大小的 AP 均提高了约 1，其中 DEIM -RT-DETRv2-M \* 取得了显著的 1.3 AP 提升。与其他方法相比，我们的方法达到了最新的最先进结果。

### 6. 附加结果

轻微修改的有效性。我们在 D-FINE-L 和 D-FINE-X 中加入了轻微修改，包括解冻 Backbone 中的 BN 层，采用 FlatCosine 学习率调度器，以及将 Decoder 激活函数替换为 SiLU。在训练了 36 个周期后，我们观察到这些变化对 D-FINE-L 没有影响，但对 D-FINE-X 来说，AP 提高了 0.1 (从 55.4 提高到 55.5)。这种配置被用作我们实验的新基准。

有无 Dense O2O 的正样本数量。在一次训练周期中，我们比较了相同训练图像在有无使用 Dense O2O 情况下的正样本数量，如图 6 所示。结合 Dense O2O 后，正样本的数量显著增加。这进一步支持了我们关于 Dense O2O 有效增强监督的论点。

我们在图 7 中展示了定性比较结果。这些结果表明，DEIM 有效地解决了 D-FINE-L 面临的两个关键问题：高置信度的重复预测和误报。例如，在顶部行，一个风筝被错误地分配了四个高度重叠的边界框，每个边界框都有高置信度评分。此外，如底部行所示，D-FINE-L 将一个插座和一个墙上物体误分类为钟表，同时未能检测瓶子。通过在训练中加入 DEIM，检测器成功解决了这些挑战。这个可视化结果突显了 DEIM 带来的显著进步，强调其在提高检测准确性方面的潜力。

Table 9. 在 COCO 上与 S 和 M 尺寸的实时目标检测器进行比较 [19] val2017。\* 表示 NMS 使用 0.01 的置信度阈值进行调优。

Model	# Epochs	# Params.	GFLOPs	Latency (ms)	AP <sup>val</sup>	AP <sub>50</sub> <sup>val</sup>	AP <sub>75</sub> <sup>val</sup>	AP <sub>S</sub> <sup>val</sup>	AP <sub>M</sub> <sup>val</sup>	AP <sub>L</sub> <sup>val</sup>
YOLO-based Real-time Object Detectors										
YOLOv8-S [12]	500	11	29	6.96	44.9	61.8	48.6	25.7	49.9	61.0
YOLOv8-M [12]	500	26	79	9.66	50.2	67.2	54.6	32.0	55.7	66.4
YOLOv9-S [33]	500	7	26	8.02	46.8	61.8	48.6	25.7	49.9	61.0
YOLOv9-M [33]	500	20	76	10.15	51.4	67.2	54.6	32.0	55.7	66.4
Gold-YOLO-S [32]	300	22	46	2.01	46.4	63.4	-	25.3	51.3	63.6
Gold-YOLO-M [32]	300	41	88	3.21	51.1	68.5	-	32.3	56.1	68.6
YOLOv10-S [31]	500	7	22	2.65	46.3	63.0	50.4	26.8	51.0	63.8
YOLOv10-M [31]	500	15	59	4.97	51.1	68.1	55.8	33.8	56.5	67.0
YOLO11-S * [13]	500	9	22	2.86	47.0	63.9	50.7	29.0	51.7	64.4
YOLO11-M * [13]	500	20	68	4.95	51.5	68.5	55.7	33.4	57.1	67.9
DETR-based Real-time Object Detectors										
RT-DETR-R18 [42]	72	20	61	4.63	46.5	63.8	50.4	28.4	49.8	63.0
RT-DETR-R34 [42]	72	31	93	6.43	48.9	66.8	52.9	30.6	52.4	66.3
RT-DETRv2-S [23]	120	20	60	4.59	48.1	65.1	57.4	36.1	57.9	70.8
DEIM-RT-DETRv2-S	120	20	60	4.59	49.0	66.1	53.3	32.6	52.5	64.1
RT-DETRv2-M [23]	120	31	92	6.40	49.9	67.5	58.6	35.8	58.6	72.1
DEIM-RT-DETRv2-M	120	31	92	6.40	50.9	68.6	55.2	34.3	54.4	67.1
RT-DETRv2-M * [23]	72	33	100	6.90	51.9	69.9	56.5	33.5	56.8	69.2
DEIM-RT-DETRv2-M *	60	33	100	6.90	53.2	71.2	57.8	35.3	57.6	70.2
D-FINE-S [26]	120	10	25	3.49	48.5	65.6	52.6	29.1	52.2	65.4
DEIM-D-FINE-S	120	10	25	3.49	49.0	65.9	53.1	30.4	52.6	65.7
D-FINE-M [26]	120	19	57	5.55	52.3	69.8	56.4	33.2	56.5	70.2
DEIM-D-FINE-M	90	19	57	5.55	52.7	70.0	57.3	35.3	56.7	69.5

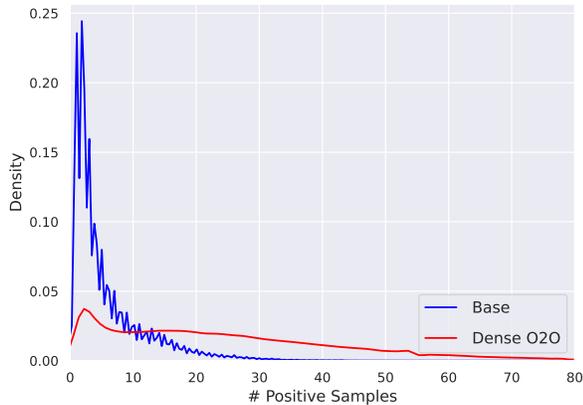


Figure 6. # 阳性样本在一个训练周期中有密集 O2O 和没有密集 O2O 的情况。Base 表示没有密集 O2O。



Figure 7. D-FINE-L 和 DEIM 之间的定性比较。在每对图像中，左边是来自 D-FINE-L 的图像，而右边是由 DEIM -D-FINE-L 预测的 (得分阈值 = 0.5)。