

YOLOv10：实时端到端物体检测

Ao Wang Hui Chen* Lihao Liu Kai Chen Zijia Lin
Jungong Han Guiguang Ding*
清华大学

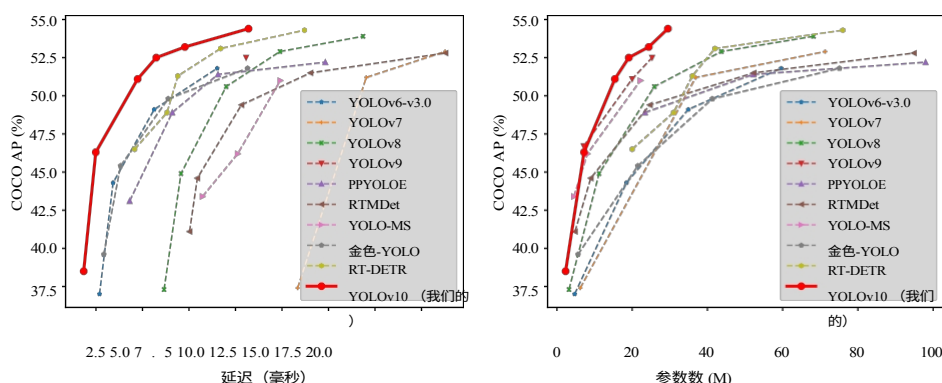


图 1：在延迟-准确性（左）和大小-准确性（右）权衡方面与其他公司的比较。我们使用官方预训练模型测量端到端延迟。

摘要

在过去几年中，YOLOs 因其在计算成本和检测性能之间的有效平衡而成为实时物体检测领域的主流模式。研究人员在 YOLO 的架构设计、优化目标、数据增强策略等方面进行了探索，取得了显著进展。然而，后处理对非最大抑制（NMS）的依赖阻碍了 YOLO 的端到端部署，并对推理延迟产生了不利影响。此外，YOLOs 中各种组件的设计缺乏全面彻底的检查，导致明显的计算冗余，限制了模型的能力。这就导致了效率不理想，性能还有很大的提升空间。在这项工作中，我们旨在从后处理和模型架构两方面进一步推进 YOLO 的性能-效率边界。为此，我们首先提出了用于 YOLOs 无 NMS 训练的一致双分配，它同时带来了有竞争力的性能和较低的推理延迟。此外，我们还为 YOLOs 引入了效率-精度驱动的整体模型设计策略。我们从效率和精度两个角度全面优化了 YOLOs 的各个组成部分，从而大大降低了计算开销，提高了能力。广泛的实验表明，YOLOv10 在各种模型尺度上都达到了最先进的性能和效率。例如，在 COCO 上的类似 AP 下，我们的 YOLOv10-S 比 RT-DETR-R18 快 1.8 倍，同时参数和 FLOPs 数量少 2.8 倍。与 YOLOv9-C 相比，在性能相同的情况下，YOLOv10-B 的延迟减少了 46%，参数减少了 25%。代码：<https://github.com/THU-MIG/yolov10>。

*通讯作者：

预印本。正在审查。

1 引言

实时物体检测一直是计算机视觉领域的研究重点，其目的是在低延迟条件下准确预测图像中物体的类别和位置。它被广泛应用于各种实际应用中，包括自动驾驶[3]、机器人导航[11]和物体跟踪[66]等。近年来，研究人员致力于设计基于 CNN 的物体检测器，以实现实时检测 [18, 22, 43, 44, 45, 51, 12]。其中，YOLOs 因其在性能和效率之间的完美平衡而越来越受欢迎[2, 19, 27, 19, 20, 59, 54, 64, 7, 65, 16, 27]。YOLOs 的检测流程由两部分组成：模型前向处理和 NMS 后处理。然而，这两部分仍存在缺陷，导致精度-延迟边界不理想。

具体来说，YOLO 在训练过程中通常采用一对多的标签分配策略，即一个地面实况对象对应多个正样本。尽管这种方法性能优越，但在推理过程中，NMS 必须选择最佳的正向预测。这会降低推理速度，使性能对 NMS 的超参数非常敏感，从而阻碍 YOLOs 实现最佳端到端部署 [71]。解决这一问题的方法之一是采用最近推出的端到端 DETR 架构 [4、74、67、28、34、40、61]。例如，RT-DETR [71] 提供了高效的混合编码器和不确定性最小的查询选择，将 DETR 推向了实时应用领域。然而，部署 DETR 本身的复杂性阻碍了它在准确性和速度之间达到最佳平衡的能力。另一条思路是探索基于 CNN 的端到端检测，通常利用一对一分配策略来抑制冗余预测 [5、49、60、73、16]。然而，这些方法通常会引入额外的推理开销，或实现次优性能。

此外，模型结构设计仍然是 YOLOs 面临的一个基本挑战，它对精度和速度有重要影响 [45, 16, 65, 7]。为了实现更高效、更有效的模型架构，研究人员探索了不同的设计策略。为提高特征提取能力，骨干网采用了多种主要计算单元，包括 DarkNet [43, 44, 45]、CSPNet [2]、EfficientRep [27] 和 ELAN [56, 58] 等。对于颈部，则探索了 PAN [35]、BiC [27]、GD [54] 和 RepGFPN [65] 等增强多尺度特征融合的方法。此外，还研究了模型缩放策略[56, 55]和重新参数化[10, 27]技术。虽然这些努力取得了显著进展，但仍缺乏从效率和准确性两个角度对 YOLOs 中各种组件的全面检测。因此，在 YOLOs 中仍然存在相当多的计算冗余，导致参数利用效率低下和效率不理想。此外，由此产生的受限模型能力也会导致性能低下，为提高精度留下了很大的空间。

在这项工作中，我们的目标是解决这些问题，并进一步推进 YOLO 的精度-速度界限。我们的目标是后处理和整个检测管道中的模型架构。为此，我们首先解决了后处理中的冗余预测问题，提出了无 NMS YOLOs 的一致双重分配策略，即双重标签分配和一致匹配度量。它使模型在训练过程中享受到丰富而和谐的监督，同时在推理过程中无需 NMS，从而获得高效的综合性能。其次，我们通过对 YOLOs 中的各个组件进行全面检查，为模型架构提出了效率-准确性驱动的整体模型设计策略。在效率方面，我们提出了轻量级分类头、空间信道解耦下采样和等级引导块设计，以减少显性计算冗余，实现更高效的架构。在精度方面，我们探索了大核卷积，并提出了有效的部分自注意模块，以增强模型能力，在低成本的情况下利用潜在的性能改进。

基于这些方法，我们成功地实现了具有不同模型尺度的新型实时端到端检测器系列，即 YOLOv10-N / S / M / B / L / X。在物体检测的标准基准（即 COCO [33]）上进行的大量实验表明，我们的 YOLOv10 在不同模型尺度上的计算量-准确度权衡方面明显优于之前的一流模型。如图 1 所示，在性能相似的情况下，我们的 YOLOv10-S / X 比 RT-DETR- R18 / R101 分别快 1.8 倍和 1.3 倍。与 YOLOv9-C 相比，YOLOv10-B 在性能相同的情况下减少了 46% 的延迟。此外，YOLOv10 还表现出高效的参数利用率。我们的 YOLOv10-L / X 比 YOLOv8-L / X 高出 0.3 个 AP，比 YOLOv8-L / X 高出 0.5 个 AP。0.5 AP，参数数量分别减少了 1.8 倍和 2.3 倍。YOLOv10-M 实现了

与 YOLOv9-M / YOLO-MS 相比, AP 与 YOLOv9-M / YOLO-MS 相似, 参数分别减少了 23% 和 31%。我们希望我们的工作能激励该领域的进一步研究和进步。

2 相关工作

实时物体检测器。实时物体检测旨在以较低的延迟对物体进行分类和定位, 这对现实世界的应用至关重要。在过去几年中, 人们一直致力于开发高效的检测器 [18, 51, 43, 32, 72, 69, 30, 29, 39]。特别是

YOLO 系列[43、44、45、2、19、27、56、20、59]是主流产品。YOLOv1、YOLOv2 和 YOLOv3 确定了由骨干、颈部和头部三部分组成的典型检测结构 [43、44、45]。YOLOv4 [2] 和 YOLOv5 [19] 引入了 CSPNet [57] 设计来取代 DarkNet [42], 同时还采用了数据增强策略、增强的 PAN 和更多的模型规模等。YOLOv6 [27] 针对颈部和骨干网分别提出了 BiC 和 SimCSPSPF, 并采用了锚点辅助训练和自蒸发策略。YOLOv7 [56] 引入了用于丰富梯度流路径的 E-ELAN, 并探索了几种可训练的自由包方法。YOLOv8 [20] 提出了用于有效特征提取和融合的 C2f 构建模块。Gold-YOLO [54] 提供了先进的 GD 机制, 以提高多尺度特征融合能力。YOLOv9 [59] 提出了改进架构的 GELAN 和增强训练过程的 PGI。

端到端物体检测器。端到端物体检测是传统管道的一种范式转变, 它提供了精简的架构 [48]。DETR [4] 引入了变压器架构, 并采用匈牙利损失来实现一对一匹配预测, 从而消除了手工制作的组件和后处理。此后, 人们提出了各种 DETR 变体, 以提高其性能和效率[40, 61, 50, 28, 34]。Deformable-DETR [74] 利用多尺度可变形注意模块来加快收敛速度。DINO [67] 在 DETR 中集成了对比去噪、混合查询选择和两次前瞻方案。RT-DETR [71] 进一步设计了高效的混合编码器, 并提出了不确定性最小的查询选择, 以提高精度和延迟。另一种实现端到端对象检测的方法是基于 CNN 的检测器。可学习的 NMS [23] 和关系网络 [25] 提出了另一种为检测器去除重复预测的网络。OneNet [49] 和 DeFCN [60] 提出了一对一匹配策略, 利用全卷积网络实现端到端物体检测。FCOS_{pss}[73] 引入了正样本选择器, 为预测选择最佳样本。

3 方法

3.1 为无 NMS 培训提供一致的双重任务分配

在训练过程中, YOLOs [20, 59, 27, 64] 通常利用 TAL [14] 为每个实例分配多个正样本。采用一对多的分配方式可以获得大量的监督信号, 从而促进优化并获得卓越的性能。但是, 这使得 YOLOs 必须依赖 NMS 后处理, 从而导致部署推理效率不理想。虽然之前的研究 [49, 60, 73, 5] 探索了一对一匹配来抑制冗余预测, 但它们通常会引入额外的推理开销或产生次优性能。在这项工作中, 我们提出了一种无 NMS 的 YOLOs 训练策略, 该策略具有双标签分配和一致的匹配度量, 可实现高效率和有竞争力的性能。

双标签分配。与一对多分配不同，一对一匹配只为每个地面实况分配一个预测，避免了 NMS 后处理。然而，这种方法会导致弱监督，从而使精度和收敛速度达不到最佳水平 [75]。幸运的是，这种缺陷可以通过一对多的分配来弥补 [5]。为此，我们为 YOLOs 引入了双标签分配，以结合两种策略的优点。具体来说，如图 2(a)所示，我们为 YOLOs 引入了另一种一对一标头。它保留了与原来一对多分支相同的结构和优化目标，但利用一对一匹配来获得标签分配。在训练过程中，两个头部与模型共同优化，使骨干和颈部享受到一对多分配提供的丰富监督。在推理过程中，我们舍弃一对多标头，利用一对一标头进行预测。这样，端到端部署就可以实现 YOLO，而不会产生任何额外的推理成本。此外，在一对一匹配中，我们采用了前一个选择，这与匈牙利匹配[4]的性能相同，但额外的训练时间更少。

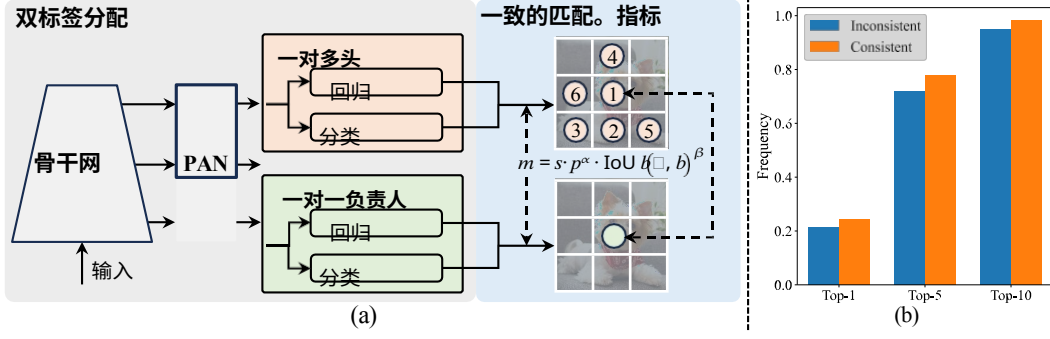


图 2: (a) 无 NMS 训练的一致双分配。(b) YOLOv8-S 默认采用 $\alpha_{o2m}=0.5$ 和 $\beta_{o2m}=6$ [20], 一对一分配在一对多结果 Top-1/5/10 中的频率。对于一致性, $\alpha_{o2o}=0.5$; $\beta_{o2o}=6$ 。对于不一致性, $\alpha_{o2o}=0.5$; $\beta_{o2o}=2$ 。

一致匹配度量。在分配过程中, 一对一和一对多方法都会利用一个指标来定量评估预测和实例之间的一致程度。为了实现两个分支的预测匹配, 我们采用了统一的匹配指标, m

$$m(\alpha, \beta) = s \cdot p^\alpha \cdot \text{IoU}(\hat{b}, b)^\beta, \quad (1)$$

其中, p 是分类得分, \hat{b} 和 b 分别表示预测和实例的边界框。 s 表示空间先验, 表示预测的锚点是否在实例内 [20, 59, 27, 64]。 α 和 β 是两个重要的超参数, 用于平衡语义预测任务和位置回归任务的影响。我们将一对多指标和一对一指标分别记为 $m_{o2m} = m(\alpha, \beta_{o2mo2m})$ 和 $m_{o2o} = m(\alpha, \beta_{o2oo2o})$ 。这些指标会影响两个头的标签分配和监督信息。

在双标签分配中, 一对多分支提供的监督信号比一对一分支丰富得多。直观地说, 如果我们能协调一对一标头和一对多标头的监督, 我们就能朝着 一对多标头的优化方向优化 一对一标头。因此, 一对一标头可以在推理过程中提高样本质量, 从而获得更好的性能。为此, 我们首先分析了两个计算头之间的监督差距。由于训练过程中的随机性, 我们在一开始就对两个 "头" 进行检查, 这两个 "头" 以相同的值初始化并产生相同的预测, 即 一对一 "头" 和 一对多 "头" 对每个预测-实例对产生相同的 p 和 IoU 。我们注意到, 两个分支的回归目标并不冲突, 因为匹配的预测会共享相同的目标, 而不匹配的预测则会被忽略。因此, 监督差距在于不同的分类目标。给定一个实例, 我们将其最 大的预测 IoU 表示为 u^* , 最大的一对多和一对多预测 IoU 表示为 u 。

一对一匹配得分分别为 m_{o2m}^* 和 m_{o2o}^* 。假设一对多分支的结果是正样本 Ω 和一对一分支选择第 i 个预测, 其指标为 $m_{o2o,i} = m_{o2o}^*$, 我们

然后可以得出分类目标 $t_j = u^* \cdot \frac{m_{o2m,j}}{m_{o2m}^*} \leq u^*$ for $j \in \Omega$ and $t_i = u^* \cdot \frac{m_{o2o,i}}{m_{o2o}^*} = u^*$

如 [20, 59, 27, 64, 14] 所述, 任务对齐损失。因此, 两个分支之间的监督差距可以通过不同分类目标的 1-Wasserstein 距离[41]得出, $\mathcal{A} = t_{o2o,i} - \mathcal{I}(i \in \Omega) t_{o2m,i}$

$$\mathcal{A} = t_{o2o,i} - \mathcal{I}(i \in \Omega) t_{o2m,i} \quad t, \quad o2m,k \quad k \in \Omega \setminus \{i\} \quad (2)$$

我们可以观察到, 差距随着 $t_{o2m,i}$ 的增大而减小, 即 i 在 Ω 中的排名靠前。当 $t_{o2m,i} = u^*$ 时, 差距达到最小, 即 i 是 Ω 中最好的正样本, 如图 2.(a) 所示。为此, 我们提出了一致匹配度量, 即 $\alpha_{o2o} = r - \alpha_{o2m}$ 和 $\beta_{o2o} = r - \beta_{o2m}$, 这意味着 $m_{o2o} = m_{o2m}^*$ 。因此, "一对多" 头部的最佳正样本也是 "一对一" 头部的最佳正样本。

因此，两个机头都可以一致、和谐地进行优化。因此，两个头部都能得到一致、和谐的优化。为简单起见，我们默认 $r=1$ ，即 $\alpha_{o2o}=\alpha_{o2m}$ 和 $\beta_{o2o}=\beta_{o2m}$ 。为了验证改进后的监督对齐情况，我们统计了训练后一对多结果的前 1 / 5 / 10 中一对一匹配对的数量。如图 2(b)所示，在一致匹配度量下，配准得到了改善。如需更全面地了解数学证明，请参阅附录。

3.2 效率-精度驱动的整体模型设计

除了后处理之外，YOLO 的模型结构也对效率和精度的权衡提出了巨大挑战[45, 7, 27]。尽管之前的工作探索了各种设计策略，但仍有许多问题需要解决、

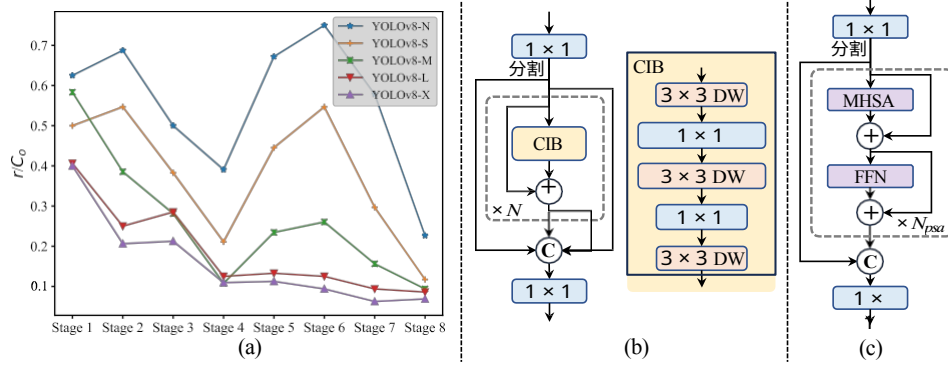


图 3: (a) YOLOv8 中各阶段和模型的内在等级。骨干和颈部的阶段按模型前进过程的顺序编号。y 轴的数值秩 r 归一化为 r/C_o ，其阈值默认设置为 $\lambda_{max}/2$ ，其中 C_o 表示输出通道数， λ_{max} 为最大奇异值。可以看出，深度阶段和大型模型表现出较低的内在秩值。(b) 紧凑型倒置块 (CIB)。(c) 部分自注意模块 (PSA)。

目前还缺乏对 YOLO 各个组件的全面检测。因此，模型结构表现出不可忽略的计算冗余和能力限制，这阻碍了其实现高效率和高性能的潜力。在此，我们旨在从效率和精度两个角度全面开展 YOLO 的模型设计。

效率驱动的设计。 YOLO 的组件包括茎干、下采样层、带有基本构件的阶段和头部。茎干的计算成本较低，因此我们对其他三个部分进行了效率驱动模型设计。

(1) **轻量级分类头。** 在 YOLO 中，分类头和回归头通常采用相同的架构。不过，它们在计算开销方面表现出明显的差异。例如，在 YOLOv8-S 中，分类头的 FLOPs 和参数数（5.95G/1.51M）分别是回归头（2.34G/0.64M）的 2.5 倍和 2.4 倍。不过，在分析了分类误差和回归误差的影响后（见表 6），我们发现回归头对 YOLO 性能的影响更大。因此，我们可以减少分类头的开销，而不必担心性能会受到很大影响。因此，我们简单地采用了一种轻量级的分类头架构，它由两个深度可分离卷积[24, 8]组成，内核大小为 3×3 ，然后是 1×1 卷积。

(2) **空间信道解耦降采样。** YOLO 通常利用步长为 2 的常规 3×3 标准卷积，实现空间降采样（从 $H \times W$ 到 $\frac{H}{2} \times \frac{W}{2}$ ）和信道降采样。

同时进行转换（从 C 到 $2C$ ）。这将带来不可忽略的计算成本 $O(\frac{9}{2} HWC^2)$ 和参数数量 $O(\frac{7}{2} 8C^2)$ 。取而代之的是，我们建议将空间缩小和信道增加操作解耦，从而实现更高效的下采样。具体来说，我们首先利用点卷积来调节信道维度，然后利用深度卷积来执行空间下采样。这将计算成本降低到 $O(2HWC^2 + 9HWC)$ ，参数数量降低到 $O(2C^2 + 18C)$ 。同时，它最大限度地保留了下采样过程中的信息，从而在降低延迟的同时实现了极具竞争力的性能。

(3) **等级引导的模块设计。** YOLO 通常在所有阶段采用相同的基本构件 [27, 59]，例如 YOLOv8 [20] 中的瓶颈构件。为了彻底检查 YOLO 的这种同质设计，我们利用本征等级 [31, 15] 来分析每个阶段的冗余度。²分析每个阶段的冗余度。具体来说，我们计算每个阶段中最后一个基本块中最后一次卷积的数值秩，它计算的是大于阈值的奇异值的数量。图 3.(a)

显示了 YOLOv8 的结果，表明深度阶段和大型模型容易出现更多冗余。这一观察结果表明，简单地对所有阶段应用相同的块设计并不能实现最佳的容量-效率权衡。为了解决这个问题，我们提出了一种等级引导的区块设计方案，旨在通过紧凑的架构设计来降低冗余阶段的复杂性。我们首先提出了一种紧凑型反转块（CIB）结构，它采用廉价的深度卷积来进行空间混合，并采用经济高效的点卷积来进行信道混合。

²等级越低，冗余越多，等级越高，信息越浓缩。

如图 3.(b) 所示。它可以作为高效的基本构件，例如嵌入 ELAN 结构[58, 20]（图 3.(b)）。然后，我们主张采用等级引导的区块分配策略，在保持有竞争力的容量的同时实现最佳效率。具体来说，给定一个模型后，我们根据其内在等级以升序对其所有阶段进行排序。我们将进一步检验用 CIB 替换领先阶段基本区块的性能变化。如果与给定模型相比性能没有下降，我们就继续替换下一阶段，反之则停止这一过程。因此，我们可以实现跨阶段和跨模型规模的自适应紧凑块设计，在不影响性能的情况下实现更高的效率。由于篇幅限制，我们在附录中提供了算法的详细信息。

精度驱动模型设计。我们进一步探索了精度驱动设计的大核卷积和自我关注，旨在以最小的成本提高性能。

(1) *大核卷积*。采用大核深度卷积是扩大感受野和增强模型能力的有效方法[9, 38, 37]。但是，如果在所有阶段都简单地利用它们，可能会对用于检测小物体的浅层特征造成污染，同时也会在高分辨率阶段带来巨大的 I/O 开销和延迟[7]。因此，我们建议在深度阶段利用 CIB 中的大核深度卷积。具体来说，我们仿照文献[37]，将 CIB 中第二个 3×3 深度卷积的核大小增加到 7×7 。此外，我们还采用了结构重参数化技术 [10, 9, 53]，引入了另一个 3×3 深度卷积分支，在不增加推理开销的情况下缓解了优化问题。此外，随着模型大小的增加，其感受野自然也会扩大，使用大核卷积的好处也会随之减少。因此，我们只在模型规模较小的情况下采用大核卷积。

(2) *部分自我注意 (PSA)*。自我注意 [52] 因其卓越的全局建模能力而被广泛应用于各种视觉任务中 [36, 13, 70]。然而，它的计算复杂度和内存占用都很高。为了解决这个问题，考虑到普遍存在的注意力冗余问题 [63]，我们提出了一种高效的部分自我注意 (PSA) 模块设计，如图 3(c)所示。具体来说，我们在 1×1 卷积后将各通道的特征平均分成两部分。我们只将一部分输入由多头自注意模块 (MHSA) 和前馈网络 (FFN) 组成的 N_{PSA} 块。然后，两部分通过 1×1 卷积进行连接和融合。此外，我们遵循文献[21]，将查询和关键字的维度分配为 MHSA 中值的一半，并用 BatchNorm [26] 代替 LayerNorm [1]，以实现快速推理。此外，PSA 只放在分辨率最低的第 4 阶段之后，避免了自我关注的二次计算复杂度带来的过高开销。这样，全局表示学习能力就能以较低的计算成本融入 YOLOs，从而很好地增强了模型的能力，提高了性能。

4 实验

4.1 实施细节

我们选择 YOLOv8 [20] 作为我们的基准模型，因为它在延迟与准确性之间达到了令人称道的平衡，而且可以提供各种规模的模型。我们采用一致的双重分配进行无 NMS 训练，并在此基础上进行效率-精度驱动的整体模型设计，这就是我们的 YOLOv10 模型。YOLOv10 具有与 YOLOv8 相同的变体，即 N / S / M / L / X。此外，我们还通过简单地增加 YOLOv10-M 的宽度比例因子推导出了新的变体 YOLOv10-B。我们在相同的从零开始训练设置下，在

COCO [33] 上验证了所提出的探测器[20, 59, 56]。此外，按照文献[71]的方法，所有模型的延迟都在使用 TensorRT FP16 的 T4 GPU 上进行了测试。

4.2 与最新技术的比较

如表 1 所示。如表 1 所示，我们的 YOLOv10 在各种模型规模下都达到了最先进的性能和端到端延迟。我们首先将 YOLOv10 与基线模型（即 YOLOv8）进行比较。在 N / S / M / L / X 五种变体上，我们的 YOLOv10 实现了 1.2% / 1.4% / 0.5% / 0.3% / 0.5% 的 AP 改进，参数减少了 28% / 36% / 41% / 44% / 57%，计算量减少了 23% / 24% / 25% / 27% / 38%，延迟降低了 70% / 65% / 50% / 41% / 37%。与其他 YOLO 相比，YOLOv10 还能在准确性和计算成本之间做出更好的权衡。具体来说，对于轻量级和小型模型，YOLOv10-N / S 的性能分别比 YOLOv6-3.0-N / S 高出 1.5 AP 和 2.0 AP。

表 1: 与最新技术的比较。延迟使用官方预训练模型进行测量。延迟_f 表示模型前向过程中的延迟, 不含后处理。† 表示 YOLOv10 使用 NMS 进行原始一对多训练的结果。为进行公平比较, 以下所有结果均未使用知识提炼或 PGI 等额外的高级训练技术。

模型	#Param.	FLOPs(G)	AP _{val} (%)	延迟 (毫秒)	延迟 _f (毫秒)
YOLOv6-3.0-N [27]	4.7	11.4	37.0	2.69	1.76
金-YOLO-N [54]	5.6	12.1	39.6	2.92	1.82
YOLOv8-N [20]	3.2	8.7	37.3	6.16	1.77
YOLOv10-N (我们的)	2.3	6.7	38.5 / 39.5[†]	1.84	1.79
YOLOv6-3.0-S [27]	18.5	45.3	44.3	3.42	2.35
Gold-YOLO-S [54]	21.5	46.0	45.4	3.82	2.73
YOLO-MS-XS [7]	4.5	17.4	43.4	8.23	2.80
YOLO-MS-S [7]	8.1	31.2	46.2	10.12	4.83
YOLOv8-S [20]	11.2	28.6	44.9	7.07	2.33
YOLOv9-S [59]	7.1	26.4	46.7	-	-
RT-DETR-R18 [71]	20.0	60.0	46.5	4.58	4.49
YOLOv10-S (我们的)	7.2	21.6	46.3 / 46.8[†]	2.49	2.39
YOLOv6-3.0-M [27]	34.9	85.8	49.1	5.63	4.56
Gold-YOLO-M [54]	41.3	87.5	49.8	6.38	5.45
YOLO-MS [7]	22.2	80.2	51.0	12.41	7.30
YOLOv8-M [20]	25.9	78.9	50.6	9.50	5.09
YOLOv9-M [59]	20.0	76.3	51.1	-	-
RT-DETR-R34 [71]	31.0	92.0	48.9	6.32	6.21
RT-DETR-R50m [71]	36.0	100.0	51.3	6.90	6.84
YOLOv10-M (我们的)	15.4	59.1	51.1 / 51.3[†]	4.74	4.63
YOLOv6-3.0-L [27]	59.6	150.7	51.8	9.02	7.90
Gold-YOLO-L [54]	75.1	151.7	51.8	10.65	9.78
YOLOv9-C [59]	25.3	102.1	52.5	10.57	6.13
YOLOv10-B (我们的)	19.1	92.0	52.5 / 52.7[†]	5.74	5.67
YOLOv8-L [20]	43.7	165.2	52.9	12.39	8.06
RT-DETR-R50 [71]	42.0	136.0	53.1	9.20	9.07
YOLOv10-L (我们的)	24.4	120.3	53.2 / 53.4[†]	7.28	7.21
YOLOv8-X [20]	68.2	257.8	53.9	16.86	12.83
RT-DETR-R101 [71]	76.0	259.0	54.3	13.71	13.58
YOLOv10-X (我们的)	29.5	160.4	54.4 / 54.4[†]	10.70	10.60

对于中型模型, 与 YOLOv9-C / YOLO-MS 相比, YOLOv10-B/M 在性能相同或更好的情况下分别减少了 46% / 62% 的延迟。对于中型模型, 与 YOLOv9-C / YOLO-MS 相比, YOLOv10-B / M 在性能相同或更好的情况下分别减少了 46% / 62% 的延迟。对于大型模型, 与 Gold-YOLO-L 相比, 我们的 YOLOv10-L 减少了 68% 的参数, 降低了 32% 的延迟, 同时显著改善了 1.4% 的 AP。此外, 与 RT-DETR 相比, YOLOv10 在性能和延迟方面都有显著提高。值得注意的是, YOLOv10-S / X 的性能和延迟分别提高了 1.8 倍和 1.5 倍。在性能相似的情况下, YOLOv10 的推理速度比 RT-DETR-R18 / R101 分别快 1.3 倍。这些结果充分证明了 YOLOv10 作为实时端到端检测器的优越性。

我们还将 YOLOv10 与其他使用原始一对多训练方法的 YOLO 进行了比较。我们按照文献 [56, 20, 54], 考虑了这种情况下模型前向过程的性能和延迟 (延迟_f)。如表 1 所示如表 1 所示, YOLOv10 在不同的模型规模下也表现出了最先进的性能和效率, 这表明我们的架构

设计是有效的。

4.3 模型分析

消融研究。表 2 列出了基于 YOLOv10-S 和 YOLOv10-M 的消融结果。2.可以看出，我们的无 NMS 训练和一致的双分配大大减少了 YOLOv10-S 的端到端延迟 4.63ms，同时保持了 44.3% AP 的竞争性能。此外，我们的效率驱动模型设计减少了 11.8 M 个参数和 20.8 GFLOPs，YOLOv10-M 的延迟时间也大幅减少了 0.65ms，充分显示了其有效性。此外，我们的精确度驱动模型设计为 YOLOv10-S 和 YOLOv10-M 分别实现了 1.8 AP 和 0.7 AP 的显著改进，而延迟开销分别仅为 0.18ms 和 0.17ms，这充分证明了其优越性。

表 2：在 COCO 上使用 YOLOv10-S 和 YOLOv10-M 进行的消融研究。

#	型号	无 NMS	。效率。	Accuracy.	#Param.(M)	FLOPs(G)	AP ^{val} (%)	Latency(ms)	
1						11.2	28.6	44.9	7.07
2	YOLOv10-S	✓				11.2	28.6	44.3	2.44
3		✓	✓			6.2	20.8	44.5	2.31
4		✓	✓	✓		7.2	21.6	46.3	2.49
5						25.9	78.9	50.6	9.50
6	YOLOv10-M	✓				25.9	78.9	50.3	5.22
7		✓	✓			14.1	58.1	50.4	4.57
8		✓	✓	✓		15.4	59.1	51.1	4.74

表 3：双重分配。

o2m	o2o	接入点延迟
✓		44.9 7.07
	✓	43.4 2.44
✓	✓	44.3 2.44

表 4：匹配度量。

α_{o2o}	β_{o2o}	AP ^{val}	α_{o2m}	β_{o2m}	AP ^{val}
0.5	2.0	42.7	0.25	3.0	44.3
0.5	4.0	44.2	0.25	6.0	43.5
0.5	6.0	44.3	1.0	6.0	43.9
0.5	8.0	44.0	1.0	12.0	44.3

表 5：YOLOv10-S/M 的效率。

# 型号	# 参数	FLOPs	AP ^{val}	延迟
1 基数	11.2/25.9	28.6/78.9	44.3/50.3	2.44/5.22
2 +cls.9.	9/23.2	23.5/67.7	44.2/50.2	2.39/5.07
3 +downs.8.0/19.7	22.2/65.0	44.4/50.4	2.36/4.97	
4 +块	6.2/14.1	20.8/58.1	44.5/50.4	2.31/4.57

对无 NMS 培训进行分析。

- **双标签分配。**我们为无 NMS YOLOs 提出了双标签分配，它既能在训练过程中为一对多（o2m）分支带来丰富的监督，又能在推理过程中为一对一（o2o）分支带来高效率。我们基于 YOLOv8-S 验证了它的优势，即表 2 中的 #1。2.具体来说，我们分别引入了只使用 o2m 分支和只使用 o2o 分支进行训练的基线。如表 3 所示如表 3 所示，我们的双标签分配实现了最佳的 AP 延迟权衡。
- **一致匹配度量。**我们引入了一致匹配度量，使一对一标头与一对多标头更加和谐。我们基于 YOLOv8-S 验证了它的优势，即表 2 中的 #1。o2oo2o如表 4 所示。如表 4 所示，所提出的一致匹配度量，即 $\alpha_{o2o} = r - \alpha_{o2m}$ 和 $\beta_{o2o} = r - \beta_{o2m}$ ，可以在一对多头[20]中达到最佳性能，其中 $\alpha_{o2m} = 0.5$ ， $\beta_{o2m} = 6.0$ 。这种改进可归因于监督间隙的减少（公式 (2)），从而改善了两个分支之间的监督一致性。此外，所提出的一致匹配度量无需进行详尽的超参数调整，这在实际应用中很有吸引力。

效率驱动模型设计分析。我们在 YOLOv10-S/M 的基础上进行实验，逐步加入效率驱动的设计元素。我们的基准是没有效率-精度驱动模型设计的 YOLOv10-S/M 模型，即表 2 中的 #2/#6。2.如表 5 所示如表 5 所示，每个设计组件，包括轻量级分类头、空间通道解耦下采样和等级引导块设计，都有助于减少参数数、FLOP 和延迟。重要的是，这些改进是在保持有竞争力的性能的同时实现的。

- **轻量级分类头。**我们以表 5 中 1 号和 2 号 YOLOv10-S 为基础，分析了预测的类别和定位误差对性能的影响。与 [6] 一样。具体来说，我们通过一对一的分配将预测与实例相匹配。那么

我们用实例标签代替预测的类别得分，结果是 AP^{val} 没有
分类错误。同样，我们用实例的位置替换预测的位置，得出
美 无 r 没有回归误差。如表 6 所示。如表 6 所示，AP^{val} 无 r 远高于 AP^{val}、
联 的估 无
社 值

结果表明，消除回归误差会带来更大的改进。性能因此，瓶颈更多地在于回归任务。因此，采用轻量级分类头可以在不影响性能的情况下提高效率。

- *空间通道解耦下采样*。为了提高效率，我们将降采样操作解耦，首先通过点卷积（PW）增加信道尺寸，然后通过深度卷积（DW）降低分辨率，以最大限度地保留信息。我们将其与基于 YOLOv10-S 的 #3（见表 5），先通过 DW 进行空间缩减，再通过 PW 进行信道调制的基线方式进行比较。5.如表 7 所示如表 7 所示，我们的下采样策略通过减少下采样过程中的信息损失，实现了 0.7% 的 AP 改进。
- *紧凑型反转块（CIB）*。我们引入 CIB 作为紧凑型基本构件。我们根据表 5 中 #4 的 YOLOv10-S 验证了它的有效性。5.具体来说，我们引入了反转残差块[46]（IRB）作为基线，它实现了 43.7% 的次优 AP，如表 8 所示。8.然后，我们在其后附加一个 3×3 深度卷积（DW），称为 "IRB-DW"，它

表 6: cls.	表 7: d.s. 的结果	表 8: CIB 的结果。	表 9: Rank-guided。
基地。	AP 型 ^{val} 基准延迟。	模型 AP ^{val} 延迟	有 CIB AP 的阶段 ^{val}
+cls. AP ^{val}	43.7 2.33	IRB 43.7 2.30	空 44.4
44.3 44.2			
美 59.9 59.9	我们的 44.4 2.3 6	IRB-DW 44.2 2.30	8 44.5
联 64.5 64.2		我们的 44.5 2.3 1	8,4, 44.5
性			
无 r			8,4,7 44.3

表 10: S/M 的精度。表 11: L.k. 结果。表 12: L.k. 使用情况。表 13: PSA 结果。

# AP 型 ^{val}	延迟	AP 型 ^{val} 延迟	w/o L.k. w/ L.k.	型号 AP ^{val} 延迟
1 个底座 44.5/50.4 2.31/4.57		k.s.=5 44.7 2.32	N36 .3 36.6	PSA 46.3 2.4 9
2 +L.k. 44.9/- 2.34/-		k.s.=7 44.9 2.3 4	S44 .5 44.9	Trans. 46.0 2.5 4
3 +PSA 46.3/51.1 2.49/4.74		K.S.=9 44.9 2.3 7	M 50.4 50.4	NPSA = 1 46.3 2.4 9
		无代表44.8 2.34		NPSA = 2 46.5 2.59

带来 0.5% 的 AP 改进。与 "IRB-DW "相比，我们的 CIB 通过预置另一个 DW，以最小的开销进一步实现了 0.3% 的 AP 改进，这表明了它的优越性。

- **秩引导块设计。**我们引入了等级引导区块设计，自适应地整合紧凑区块设计，以提高模型效率。我们根据表 5 中 #3 的 YOLOv10-S 验证了它的益处。5.根据内在等级从高到低排序的阶段为阶段 8-4-7-3-5-1-6-2，如图 3(a)。如表 9 所示如表 9 所示，当用高效 CIB 逐步替换每个阶段的瓶颈块时，我们发现从阶段 7 开始性能下降。在第 8 和第 4 阶段，由于内在等级较低、冗余较多，我们可以在不影响性能的情况下采用高效区块设计。这些结果表明，等级引导区块设计可以作为提高模型效率的有效策略。

精度驱动模型设计分析。我们介绍了在 YOLOv10-S/M 基础上逐步整合精度驱动设计元素的结果。我们的基线是纳入效率驱动设计后的 YOLOv10-S/M 模型，即表 2 中的 #3/#7。2.如表 10 所示如表 10 所示，采用大内核卷积和 PSA 模块后，YOLOv10-S 的性能在延迟增加 0.03 毫秒和 0.15 毫秒的情况下分别提高了 0.4% AP 和 1.4% AP。请注意，YOLOv10-M 没有使用大核卷积（见表 12）。

- **大核卷积**我们首先根据表 10 中 #2 的 YOLOv10-S 研究了不同核大小的影响。10.如表 11 所示如表 11 所示，随着内核大小的增大，性能有所提高，在内核大小为 7×7 时性能停滞不前，这说明了大感知场的好处。此外，在训练过程中去掉重新参数化分支后，AP 下降了 0.1%，显示了其优化效果。此外，我们以 YOLOv10-N / S / M 为基础，考察了大核卷积在不同模型规模下的优势。如表 12 所示，由于 YOLOv10-M 固有的广泛感受野，它对大型模型（即 YOLOv10-M）没有任何改进。因此，我们只对小型模型（即 YOLOv10-N / S）采用大核卷积。
- **部分自我关注 (PSA)。**我们引入了 PSA，以最小的成本结合全局建模能力来提高性能。我们首先根据表 10 中 #3 的 YOLOv10-S 验证了 PSA 的有效性。10.具体来说，我们引入变压器模块，即 MHSA 后接 FFN 作为基线，记为 "Trans"。如表 13 所示13 所示，与之相比，PSA 的 AP 提高了 0.3%，延迟减少了 0.05ms。性能提升的原因可能是通过减少注意

力头的冗余，缓解了自我注意力的优化问题 [62, 9]。此外，我们还研究了不同 N_{PSA} 的影响。如表 13 所示，将 N_{PSA} 增加到 2 时，AP 提高了 0.2%，但延迟增加了 0.1ms。因此，我们将 N_{PSA} 默认设置为 1，以在保持高效率的同时增强模型能力。

5 结论

在本文中，我们将后处理和模型架构作为整个 YOLOs 检测流水线的目标。在后处理方面，我们提出了无 NMS 训练的一致双重分配，实现了高效的端到端检测。在模型架构方面，我们引入了效率-精度驱动的整体模型设计策略，改善了性能-效率权衡。这为我们带来了全新的实时端到端对象检测器 YOLOv10。广泛的实验表明，与其他先进的检测器相比，YOLOv10 的性能和延迟都达到了最先进的水平，充分证明了它的优越性。

参考资料

- [1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton.《层归一化》, *arXiv preprint arXiv:1607.06450*, 2016.
- [2] Alexey Bochkovskiy、Chien-Yao Wang 和 Hong-Yuan Mark Liao。Yolov4: 物体检测的最佳速度和精度, 2020年。
- [3] Daniel Bogdoll、Maximilian Nitsche 和 J Marius Zöllner.自动驾驶中的异常检测: 调查。 *IEEE/CVF 计算机视觉与模式识别会议论文集* , 第 4488-4499 页, 2022 年。
- [4] Nicolas Carion、Francisco Massa、Gabriel Synnaeve、Nicolas Usunier、Alexander Kirillov 和 Sergey Zagoruyko。使用变换器进行端到端物体检测。 *欧洲计算机视觉会议* , 第 213-229 页。Springer, 2020.
- [5] Yiqun Chen、Qiang Chen、Qinghao Hu 和 Jian Cheng。日期: 端到端全卷积对象检测的双重分配。 *arXiv preprint arXiv:2211.13859*, 2022.
- [6] Yiqun Chen, Qiang Chen, Peize Sun, Shoufa Chen, Jingdong Wang, and Jian Cheng.《用方框细化增强训练有素的检测器》, *arXiv 预印本 arXiv:2307.11828*, 2023.
- [7] 陈玉明、袁新斌、吴瑞琪、王家宝、侯启斌、程明明。Yolo-ms: Rethinking multi-scale representation learning for real-time object detection. *ArXiv preprint arXiv:2308.05480*, 2023.
- [8] 弗朗索瓦-乔莱Xception: 深度可分离卷积深度学习。In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251-1258, 2017.
- [9] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding.将内核扩大到 31x31: 重新审视 cnns 中的大内核设计。 *IEEE/CVF 计算机视觉与模式识别会议论文集* , 第 11963-11975 页, 2022 年。
- [10] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun.Repvgg: 让 Vgg 风格的 Convnets 再次伟大。 *IEEE/CVF 计算机视觉与模式识别会议论文集* , 第 13733-13742 页, 2021 年。
- [11] Douglas Henke Dos Reis、Daniel Welfer、Marco Antonio De Souza Leite Cuadros 和 Daniel Fernando Tello Gamarra。使用rgb图像和yolo算法的物体识别软件进行移动机器人导航。 *应用人工智能* , 33 (14) : 1290-1305, 2019.
- [12] 段凯文、白松、谢灵茜、齐红刚、黄清明、田琦。Centernet: 用于物体检测的关键点三元组。In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6569-6578, 2019.
- [13] Patrick Esser、Robin Rombach 和 Bjorn Ommer。驯服变换器, 实现高分辨率图像合成。 *IEEE/CVF 计算机视觉与模式识别会议论文集* , 第 12873-12883 页, 2021 年。

- [14] Chengjian Feng, Yujie Zhong, Yu Gao, Matthew R Scott, and Weilin Huang. Tood: 任务对齐的单级物体检测。2021 年 *IEEE/CVF 计算机视觉国际会议 (ICCV)* , 第 3490-3499 页。IEEE 计算机协会, 2021 年。
- [15] 冯瑞丽、郑克成、黄玉昆、赵德利、迈克尔-乔丹和查正军。深度神经网络中的等级递减。《*神经信息处理系统进展*》, 35:33054-33065, 2022 年。
- [16] Zheng Ge、Songtao Liu、Feng Wang、Zeming Li 和 Jian Sun. YoloX: *ArXiv preprint arXiv:2107.08430*, 2021.
- [17] Golnaz Ghiasi、Yin Cui、Aravind Srinivas、Rui Qian、Tsung-Yi Lin、Ekin D Cubuk、Quoc V Le 和 Barret Zoph。简单复制粘贴是实例分割的强大数据增强方法。《*IEEE/CVF 计算机视觉与模式识别会议论文集*》, 第 2918-2928 页, 2021 年。

- [18] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440-1448, 2015.
- [19] Jocher Glenn. <https://github.com/ultralytics/yolov5/tree/v7.0>, 2022.
- [20] Jocher Glenn. <https://github.com/ultralytics/ultralytics/tree/main>, 2023.
- [21] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: 披着 convnet 外衣的视觉转换器, 实现更快推理。 In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12259-12269, 2021.
- [22] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961-2969, 2017.
- [23] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. 学习非最大抑制。《电气和电子工程师学会计算机视觉与模式识别会议论文集》, 第 4507-4515 页, 2017 年。
- [24] 安德鲁-G-霍华德、朱梦龙、陈波、德米特里-卡连琴科、王伟军、托比亚斯-韦扬德、马可-安德烈托和哈特维格-亚当。移动网络: 用于移动视觉应用的高效卷积神经网络。 *arXiv preprint arXiv:1704.04861*, 2017.
- [25] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. 用于物体检测的关系网络。 In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3588-3597, 2018.
- [26] Sergey Ioffe 和 Christian Szegedy. 批量归一化: 通过减少内部协变量偏移加速深度网络训练。 pmlr, 2015 年。
- [27] 李楚怡、李璐璐、耿一飞、蒋洪亮、程萌、张博、柯在丹、徐晓明、储祥祥。 Yolov6 v3.0: *ArXiv preprint arXiv:2301.05586*, 2023.
- [28] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: 通过引入查询去噪加速检测训练。 *IEEE/CVF 计算机视觉与模式识别会议论文集》*, 第 13619-13627 页, 2022 年。
- [29] Xiang Li, Wenhai Wang, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. 广义焦点损失 v2: 为密集物体检测学习可靠的定位质量估计。 *IEEE/CVF 计算机视觉与模式识别会议论文集》*, 第 11632-11641 页, 2021 年。
- [30] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. 广义焦点损失: 为密集物体检测学习合格的分布式边界框 *神经信息处理系统进展》*, 33:21002-21012, 2020.
- [31] Ming Lin, Hesun Chen, Xiuyu Sun, Qi Qian, Hao Li, and Rong Jin. GPU 高效网络的神经架构设计. *ARXiv 预印本 arXiv:2006.14090*, 2020.

- [32] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár.密集物体检测的焦点丢失。 In *Proceedings of the IEEE international conference on computer vision*, pages 2980-2988, 2017.
- [33] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick.微软 coco：上下文中的常见对象。 *计算机视觉-ECCV 2014：第13届欧洲会议，瑞士苏黎世，2014年9月6-12日，论文集，第V13部分*，第740-755页。 Springer, 2014.
- [34] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang.Dab-detr： *ArXiv preprint arXiv:2201.12329*, 2022.

- [35] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. 用于实例分割的路径聚合网络。 In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759-8768, 2018.
- [36] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin 和 Baining Guo. Swin 变换器：使用移位窗口的分层视觉变换器。 *IEEE/CVF 计算机视觉国际会议论文集*，第 10012-10022 页，2021 年。
- [37] 刘壮、毛汉子、吴超元、克里斯托夫-费希滕霍夫、特雷弗-达雷尔和谢赛宁。面向 2020 年代的 convnet。 *IEEE/CVF 计算机视觉与模式识别会议论文集*，第 11976-11986 页，2022 年。
- [38] 罗文杰、李宇佳、拉奎尔-乌塔松和理查德-泽梅尔。理解深度卷积神经网络中的有效感受野。 *神经信息处理系统进展*，2016 年第 29 期。
- [39] 柳承启、张文伟、黄海安、周玥、王玉东、刘彦怡、张世龙、陈凯。 Rtmnet: *ARXiv 预印本 arXiv:2212.07784*, 2022.
- [40] 孟 德普、陈小康、范泽佳、曾刚、李厚强、袁宇辉、孙磊、王景东。快速训练收敛的条件检测。 *IEEE/CVF 计算机视觉国际会议论文集*，第 3651-3660 页，2021 年。
- [41] Victor M Panaretos 和 Yoav Zemel. 沃瑟斯坦距离的统计方面。 *统计及其应用年度综述*，6:405-431, 2019.
- [42] 约瑟夫-雷德蒙 Darknet: <http://pjreddie.com/darknet/>, 2013-2016。
- [43] Joseph Redmon, Santosh Divvala, Ross Girshick 和 Ali Farhadi. 只看一次统一的实时物体检测。 *电气和电子工程师学会计算机视觉与模式识别会议 (CVPR) 论文集*, 2016 年 6 月。
- [44] 约瑟夫-雷德蒙和阿里-法哈迪 Yolo9000: 更好、更快、更强。 *电气和电子工程师协会计算机视觉与模式识别大会 (CVPR) 论文集*, 2017 年 7 月。
- [45] 约瑟夫-雷德蒙和阿里-法哈迪 Yolo v3: 渐进式改进, 2018 年。
- [46] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov 和 Liang-Chieh Chen. Mobilenetv2: 倒残差与线性瓶颈。 In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510-4520, 2018.
- [47] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: 用于物体检测的大规模高质量数据集。 In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430-8439, 2019.
- [48] Russell Stewart, Mykhaylo Andriluka 和 Andrew Y Ng. 拥挤场景中的端到端人员检测。 *电气和电子工程师学会计算机视觉与模式识别会议论文集*，第 2325-2333 页，2016 年。

- [49] 孙培泽、蒋毅、谢恩泽、邵文琪、袁泽环、王长虎、罗平。端到端物体检测靠什么？
国际机器学习大会，第 9934-9944 页。PMLR, 2021.
- [50] Peize Sun、Rufeng Zhang、Yi Jiang、Tao Kong、Chenfeng Xu、Wei Zhan、Masayoshi Tomizuka、Lei Li、Zehuan Yuan、Changhu Wang 等 Sparse r-cnn: End-to-end object detection with learnable proposals.*IEEE/CVF 计算机视觉与模式识别会议论文集*，第 14454-14463 页，2021 年。
- [51] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He.Fcos: 简单而强大的无锚对象检测器。
电气与电子工程师学会模式分析与机器智能论文集，44 (4) : 1922-1933, 2020 年。

- [52] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Łukasz Kaiser 和 Illia Polosukhin。注意力就是你所需要的一切。《*神经信息处理系统进展*》，2017年30期。
- [53] 王敖、陈辉、林子嘉、蒲恒军、丁贵光。Repvit: *ArXiv preprint arXiv:2307.09283*, 2023.
- [54] 王程程、何伟、聂颖、郭建元、刘传健、王云鹤、韩凯。Gold-yolo: 通过聚散机制的高效物体检测器。《*神经信息处理系统进展*》，36, 2024。
- [55] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao.Scaled-yolov4: Scaling cross stage partial network.《*IEEE/cvf 计算机视觉与模式识别会议论文集*》，第 13029-13038 页，2021 年。
- [56] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao.Yolov7: 可训练的免费包为实时物体检测器树立了新的标杆。《*IEEE/CVF 计算机视觉与模式识别会议论文集*》，第 7464-7475 页，2023 年。
- [57] Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, and I-Hau Yeh.Cspnet: 可增强 cnn 学习能力的新骨干网。《*IEEE/CVF 计算机视觉与模式识别研讨会论文集*》，第 390-391 页，2020 年。
- [58] Chien-Yao Wang, Hong-Yuan Mark Liao, and I-Hau Yeh.通过梯度路径分析设计网络设计策略》，*arXiv preprint arXiv:2211.04800*, 2022.
- [59] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao.Yolov9: Learning what you want to learn using programmable gradient information. *ArXiv preprint arXiv:2402.13616*, 2024.
- [60] 王剑锋、宋林、李泽明、孙宏斌、孙健和郑南宁。全卷积网络的端到端物体检测。《*IEEE/CVF 计算机视觉与模式识别会议论文集*》，第 15849-15858 页，2021 年。
- [61] 王英明、张翔宇、杨彤、孙健。锚点检测器：基于变压器的检测器的查询设计。In *Proceedings of the AAAI conference on artificial intelligence, volume 36*, pages 2567-2575, 2022.
- [62] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang.Cvt: 将卷积引入视觉变换器。《*IEEE/CVF 计算机视觉国际会议论文集*》，第 22-31 页，2021 年。
- [63] 徐海洋、周志超、何东亮、李富、王敬东。具有注意图幻觉和ffn压缩的视觉变换器。*arXiv预印本arXiv:2306.10875*, 2023。
- [64] Pp-yoloe: *ArXiv preprint arXiv:2203.16250*, 2022.
- [65] Xianzhe Xu, Yiqi Jiang, Weihua Chen, Yilun Huang, Yuan Zhang, and Xiuyu Sun.Damo-yolo: 实时物体检测设计报告。 *arXiv preprint arXiv:2211.15444*, 2022.

- [66] 曾凡高、董斌、张宇昂、王天财、张翔宇、魏一辰。Motr：使用变压器的端到端多目标跟踪。《欧洲计算机视觉会议》，第 659-675 页。Springer, 2022.
- [67] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino：用于端到端物体检测的带改进去噪锚盒的 Detr。 *arXiv 预印本 arXiv:2203.03605*, 2022.
- [68] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz： *ArXiv preprint arXiv:1710.09412*, 2017.

- [69] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li.通过自适应训练样本选择缩小基于锚和无锚检测之间的差距*IEEE/CVF 计算机视觉与模式识别会议论文集*，第 9759-9768 页，2020 年。
- [70] 张文强、黄子龙、罗国忠、陈涛、王兴刚、刘文宇、于刚、沈春华。Topformer：用于移动语义分割的令牌金字塔变换器。*IEEE/CVF 计算机视觉与模式识别大会论文集*，第 12083-12093 页，2022 年。
- [71] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen.Detr在实时物体检测上击败Yolos。*arXiv 预印本arXiv:2304.08069*，2023。
- [72] 郑朝晖、王平、刘伟、李金泽、叶荣光、任东伟。距离损耗：更快更好的边界框回归学习。In *Proceedings of the AAAI conference on artificial intelligence, volume 34*, pages 12993-13000, 2020.
- [73] Qiang Zhou 和 Chaohui Yu.通过消除启发式 nms 简化物体检测*电气和电子工程师学会多媒体论文集*，2023 年。
- [74] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai.可变形检测器：用于端到端对象检测的可变形变换器。*arXiv 预印本 arXiv:2010.04159*, 2020.
- [75] 宗卓凡、宋光禄、刘宇。使用协作混合任务训练的 Detrs。*IEEE/CVF 计算机视觉国际会议论文集*，第 6748-6758 页，2023 年。

A 附录

A.1 实施细节

按照文献[20, 56, 59], 所有 YOLOv10 模型都使用 SGD 优化器从头开始训练 500 次。SGD 动量和权重衰减分别设置为 0.937 和 5×10^{-4} 。初始学习率为 1×10^{-2} , 然后线性衰减至 1×10^{-4} 。在数据扩增方面, 我们采用了 Mosaic [2, 19]、Mixup [68] 和复制粘贴扩增 [17] 等方法, 如 [20, 59]。表 14 列出了详细的超参数。所有模型都是在 8 个英伟达 3090 GPU 上训练的。此外, 我们将 YOLOv10-M 的宽度比例因子提高到 1.0, 从而得到 YOLOv10-B。对于 PSA, 我们采用了 SPPF 模块[20], 并对 FFN 采用了 2 的扩展因子。对于 CIB, 我们也对倒置瓶颈块结构采用 2 的扩展率。按照文献[59, 56], 我们报告了 COCO 数据集[33]中不同对象尺度和 IoU 阈值下的标准平均精度 (AP)。

此外, 我们按照文献[71]建立了端到端速度基准。由于 NMS 的执行时间受输入的影响, 因此我们像 [71] 一样, 测量 COCO val set 上的延迟。我们采用的 NMS 超参数与检测器在验证过程中使用的相同。TensorRT efficientNMSPlugin 被附加用于后处理, I/O 开销被省略。我们报告了所有图像的平均延迟。

表 14: YOLOv10 的超参数。

超参数	YOLOv10-N/S/M/B/L/X
纪元	500
优化器	SGD
动量	0.937
重量衰减	5×10^{-4}
热身时间	3
热身动力	0.8
热身偏差学习率	0.1
初始学习率	10^{-2}
最终学习率	10^{-4}
学习率表	线性衰变
箱内损失收益	7.5
班级损失收益	0.5
DFL 损失收益	1.5
HSV 饱和度增强	0.7
增加 HSV 值	0.4
HSV 色调增强	0.015
翻译扩增	0.1
扩大规模	0.5/0.5/0.9/0.9/0.9/0.9
马赛克扩增	1.0
混合增强	0.0/0.0/0.1/0.1/0.15/0.15
复制粘贴扩增	0.0/0.0/0.1/0.1/0.3/0.3
闭合镶嵌纪元	10

A.2 一致匹配度量标准的详情

我们在此提供一致匹配度量的详细推导。

如本文所述, 我们假设一对多正样本为 Ω , 一对一支选择第 i 个预测。然后, 我们可以利用归一化指标 [14] 得到

任务对齐学习的分类目标 [20、14、59、27、64]，即 $t_{o2m,j} = u \cdot \frac{m_{o2m,j}}{m_{o2m}^*} \leq u^*$

为 $j \in \Omega$ 和 $t_{o2o,i} = u \cdot \frac{m_{o2o,i}}{m_{o2o}^*} = u^*$ 。因此，我们可以推导出两个

通过不同分类目标的 1-Wasserstein 距离[41]，即..、 \sum

$$\begin{aligned} A &= |(1 - t_{o2o,i}) - (1 - I(i \in \Omega)t_{o2m,i})| + \sum_{k \in \Omega \setminus \{i\}} |1 - (1 - t_{o2m,k})| \\ &= |t_{o2o,i} - I(i \in \Omega)t_{o2m,i}| + \sum_{k \in \Omega \setminus \{i\}} t_{o2m,k} \\ &= t_{o2o,i} - I(i \in \Omega)t_{o2m,i} + \sum_{k \in \Omega \setminus \{i\}} t_{o2m,k} \end{aligned} \quad (3)$$

其中, $I(-)$ 是指标函数。我们将 Ω 中预测的分类目标表示为 $\{\hat{t}_1, \hat{t}_2, \dots, \hat{t}_{|\Omega|}\}$ 依次递减, $\hat{t}_1 \geq \hat{t}_2 \geq \dots \geq \hat{t}_{|\Omega|}$ 。然后, 我们可以用 u 替换 $t_{o2o,i}^*$ 并获得

$$\begin{aligned} A &= u^* - I(i \in \Omega) t_{o2m,i} + \sum_{k \in \Omega \setminus \{i\}} t_{o2m,k} - 2 - I(i \in \Omega) t_{o2m,i} \\ &= u^* + \sum_{k \in \Omega} t_{o2m,k} - 2 - I(i \in \Omega) t_{o2m,i} \\ &= u^* + \sum_{k=1}^{|\Omega|} \hat{t}_k - 2 - I(i \in \Omega) t_{o2m,i} \end{aligned} \quad (4)$$

我们进一步讨论了两种情况下的监管差距, 即

1. 假设 $i \notin \Omega$, 我们可以得到

$$A = u^* + \sum_{k=1}^{|\Omega|} \hat{t}_k \quad (5)$$

2. 假设 $i \in \Omega$, 我们表示 $t_{o2m,i} = \hat{t}_n$, 得到

$$A = u^* + \sum_{k=1}^{|\Omega|} \hat{t}_k - 2 - \hat{t}_n \quad (6)$$

由于 $\hat{t}_n \geq 0$, 第二种情况会导致较小的监督差距。此外, 我们还可以观察到, 随着 \hat{t}_n 的增大, A 会减小, 这表明 n 会减小, i 在 Ω 内的排名会提升。由于 $\hat{t}_n \leq \hat{t}_1$, 当 $\hat{t}_n = \hat{t}_1$ 时, A 达最小值, 即 i 是 Ω 中最好正向样本, 且 $\hat{t} = \hat{t}^*$ 。
 $m_{o2m,i} = m_{o2m}^*$ 和 $t_{o2m,i} = u^* - \frac{m_{o2m,i}^*}{m_{o2m}^*} u^* = u^*$ 。

此外, 我们还证明可以通过一致匹配度量实现最小的监督差距。我们假设 $\alpha_{o2m} > 0$ 和 $\beta_{o2m} > 0$,

这在 [20, 59, 27, 14, 64] 中很常见。类似地、

我们假设 $\alpha_{o2o} > 0$ 和 $\beta_{o2o} > 0$ 。我们可以得到 $r_1 = \frac{\alpha_{o2o}}{\alpha_{o2m}} > 0$ 和 $r_2 = \frac{\beta_{o2o}}{\beta_{o2m}} > 0$, 那么

得出 m_{o2o}

$$\begin{aligned} m_{o2o} &= s - p^{\alpha_{o2o}} - \text{IoU}(\hat{b}, b)^{\beta_{o2o}} \\ &= s - p^{r_1 \cdot \alpha_{o2m}} - \text{IoU}(\hat{b}, b)^{r_2 \cdot \beta_{o2m}} \\ &= s - (p^{\alpha_{o2m}} - \text{IoU}(\hat{b}, b)^{\beta_{o2m} r_1} - \text{IoU}(\hat{b}, b)^{(r_2 - r_1) \beta_{o2m}} \\ &= m_{o2m}^{r_1} - \text{IoU}(\hat{b}, b)^{(r_2 - r_1) \beta_{o2m}} \end{aligned} \quad (7)$$

要实现 $m_{o2m,i} = m_{o2m}^*$ 和 $m_{o2o,i} = m_{o2o}^*$ 因此, 我们可以使 m_{o2o} 随下列因素单调递增 m_{o2m} , 即指定 $(r_2 - r_1) = 0$ 、

$$m_{o2o} = m_{o2m}^{r_1} - \text{IoU}(\hat{b}, b)^{0 \cdot \beta_{o2m}} = m_{o2m}^{r_1} \quad (8)$$

假设 $r_1 = r_2 = r$, 我们可以得出一致匹配度量, 即 $\alpha_{o2o} = r - \alpha_{o2m}$ 和

$\beta_{o2o} = r - \beta_{o2m}$ 。只需取 $r = 1$, 就可以得到 $\alpha_{o2o} = \alpha_{o2m}$ 和 $\beta_{o2o} = \beta_{o2m}$

A.3 梯级引导区块设计的详细信息

我们将详细介绍 Algo 中的秩引导区块设计算法。1. 此外, 为了计算卷积的数值秩, 我们将其权重重塑为 $(C_o, K^2 \times C_i)$ 的形状, 其中 C_o 和 C_i 分别表示输出和输入通道的数量, K 表示内核大小。

A.4 关于 COCO 的更多结果

我们报告了 YOLOv10 在 COCO (包括 AP^{val} 和 AP^{val}) 上的详细性能。

IoU 临界值以及 AP^{val} 表 15 列出了不同尺度上的 "阈值"。15.

小, 美
联
社

AP^{val}
medium

A.5 以效率-精度为导向的整体模型设计的更多分析

我们注意到，由于 YOLOv10-S（表 2 中的 #2）的模型规模较小，减少其延迟时间尤其具有挑战性。不过，如表 2 所示，我们的效率驱动型模型设计仍然实现了 5.3% 的时延缩减，而不需要在模型规模上进行调整。2 中所示，我们的效率驱动模型设计仍然在不影响性能的情况下将延迟时间减少了 5.3%。这为进一步的精度驱动模型设计提供了有力支持。YOLOv10-S 通过我们的效率-准确度整体驱动模型设计，实现了更好的延迟-准确度权衡，显示出 2.0% 的 AP 改进，而仅有

算法 1: 梯度引导区块设计

输入: 所有阶段 S 的内在等级 R ; 原始网络 Θ ; CIB θ_{cib}

输出: 新网络 Θ^* 与某些阶段的 CIB。

```

1  $t \leftarrow 0$ ;
2  $\Theta_0 \leftarrow \Theta$ ;  $\theta^* \leftarrow \theta_0$ ;
3  $ap_0 \leftarrow AP(T(\Theta_0))$ ; //  $T$ : 训练网络;  $AP$ : 评估  $AP$  性能。
4 当  $S \neq \emptyset$  时做
5    $st \leftarrow \operatorname{argmin}_{s \in S} R$ ;
6    $\Theta_{t+1} \leftarrow \operatorname{Replace}(\Theta_t, \theta_{cib}, \mathbf{s}_t)$ ; // 用 CIB  $\theta_{cib}$  替换  $\Theta_t$  阶段  $\mathbf{s}_t$  中的块。
7    $ap_{t+1} \leftarrow AP(T(\Theta_{t+1}))$ ;
8   如果  $ap_{t+1} \geq ap_0$ , 那么
9      $\Theta^* \leftarrow \Theta_{t+1}$ ;  $S \leftarrow S \setminus \{st\}$ ;
10  不然
11    返回  $\Theta$ ; *
12  最后
13 结束
14 return  $\Theta^*$ ;

```

表 15: COCO 上 YOLOv10 的详细性能。

模型	AP^{val} (%)	AP_{50}^{val} (%)	AP_{75}^{val} (%)	AP_{A}^{val} (%)	$AP_{中等}^{val}$ (%)	$AP_{大}^{val}$ (%)
YOLOv10-N	38.5	53.8	41.7	18.9	42.4	54.6
YOLOv10-S	46.3	63.0	50.4	26.8	51.0	63.8
YOLOv10-M	51.1	68.1	55.8	33.8	56.5	67.0
YOLOv10-B	52.5	69.6	57.2	35.1	57.8	68.5
YOLOv10-L	53.2	70.1	58.1	35.8	58.5	69.4
YOLOv10-X	54.4	71.3	59.3	37.0	59.8	70.9

0.05ms 的延迟开销。此外，对于模型规模更大、冗余度更高的 YOLOv10-M（表 2 中的 #6），我们的效率驱动模型设计大大减少了 12.5% 的延迟，如表 2 所示。2. 当结合精度驱动模型设计时，我们观察到 YOLOv10-M 的 AP 显著提高了 0.8%，同时延迟减少了 0.48ms。这些结果充分证明了我们的设计策略在不同模型规模下的有效性。

A.6 可视化结果

图 4 展示了我们的 YOLOv10 在复杂和具有挑战性的场景中的可视化结果。可以看出，YOLOv10 可以在各种困难条件下实现精确检测，如弱光、旋转等。此外，YOLOv10 在检测瓶子、杯子和人等各种密集物体方面也表现出很强的能力。这些结果表明其性能优越。

A.7 贡献、局限性和更广泛的影响

贡献。 总之，我们的贡献有以下三个方面：

1. 我们为无 NMS YOLOs 提出了一种新颖的一致双赋值策略。我们设计了一种双标签分配方式，在训练过程中通过一对多分支提供丰富的监督，在推理过程中通过一对一分支提供高效的监督。此外，为了确保两个分支之间的和谐监督，我们创新性地提出了一致匹配度量，从而很好地缩小了理论监督差距，提高了性能。

2. 我们针对 YOLO 的模型架构提出了一种效率-精度驱动的整体模型设计策略。我们提出了新颖的轻量级分类头、空间信道解耦下采样和秩引导块设计，大大减少了计算冗余，实现了高效率。我们进一步引入了大核卷积和创新的部分自注意模块，在低成本的情况下有效提高了性能。
3. 基于上述方法，我们推出了新型实时端到端物体检测器 YOLOv10。广泛的实验证明，与其他先进的检测器相比，我们的 YOLOv10 在性能和效率权衡方面都达到了最先进的水平。

