YOLO-World：实时开放词汇对象检测

YOLO 系列检测器已被公认为高效实用的工具。然而，它们依赖于预定义和训练好的目标类别，限制了其在开放场景中的适用性。为了解决这一局限性，我们推出了 YOLO-World，这是一种创新方法，通过视觉语言建模和在大规模数据集上进行预训练，增强了 YOLO 的开放词汇检测能力。具体而言，我们提出了一种新的可重新参数化的视觉语言路径聚合网络 (RepVL-PAN) 和区域文本对比损失函数，以促进视觉和语言信息之间的交互。我们的方法能够高效地以零样本方式检测各种目标。在极具挑战性的 LVIS 数据集上，YOLO-World 在 V100 上实现了 35.4 AP 和 52.0 FPS，在准确率和速度方面均超越了许多最先进的方法。此外，经过微调的 YOLO-World 在多个下游任务中也取得了卓越的性能，包括目标检测和开放词汇实例分割。

1. 引言

目标检测一直是计算机视觉领域中一个长期存在且至关重要的挑战，在图像理解、机器人技术和自动驾驶汽车领域有着广泛的应用。随着深度神经网络的发展，大量的研究 [17, 27, 43, 45] 在目标检测方面取得了重大突破。尽管这些方法取得了成功，但它们仍然受到限制，因为它们只能检测具有固定词汇的目标，例如 COCO [26] 数据集中的 80 个类别。一旦定义并标记了目标类别，经过训练的检测器就只能检测这些特定的类别，从而限制了其在开放场景下的能力和适用性。

最近的研究 [9, 14, 48, 53, 59] 探索了流行的视觉语言模型 [19, 39]，通过从语言编码器中提取词汇知识来解决开放词汇检测 [59]，例如 BERT [6]。然而，这些基于知识蒸馏的方法由于训练数据稀缺且词汇多样性有限而受到很大限制，例如包含 48 个基本类别的 OV-COCO [59]。一些方法 [24, 30, 56, 57, 60] 将目标检测训练重新表述为区域级视觉语言预训练，并大规模训练开放词汇目标检测器。然而，这些方法在实际场景中仍然难以实现检测，主要受到两个方面的影响：（1）计算负担重；（2）边缘设备的部署复杂。先前的研究[24, 30, 56, 57, 60]已经证明了预训练大型检测器的良好性能，而预训练小型检测器以赋予其开放识别能力的方法尚待探索。

本文提出了YOLO-World，旨在实现高效的开放词汇目标检测，并探索了大规模预训练方案，以将传统的YOLO检测器提升到新的开放词汇领域。与以往的方法相比，YOLOWorld效率更高，推理速度更快，并且易于部署到下游应用中。具体而言，YOLO-World遵循标准YOLO架构[20]，并利用预训练的CLIP [39]文本编码器对输入文本进行编码。我们进一步提出了可重参数化的视觉语言路径聚合网络（RepVL-PAN），以连接文本特征和图像特征，从而获得更好的视觉语义表示。在推理过程中，可以移除文本编码器，并将文本向量重新参数化为 RepVLPAN 的权重，以实现高效部署。我们进一步研究了 YOLO 检测器的预训练方案，该方案基于大规模数据集进行区域文本对比学习，将检测、基础数据和图文数据统一为区域文本对。预训练的 YOLO-World 拥有丰富的区域文本对，展现出强大的开放词汇检测能力，并且训练更多数据可以带来更大的开放词汇能力提升。

此外，我们探索了一种“提示-检测”范式，以进一步提升开放词汇目标检测在实际场景中的效率。如图 2 所示，传统的目标检测器 [17, 20, 23, 41–43, 52] 专注于使用预定义和训练类别进行固定词汇（闭集）检测。而之前的开放词汇检测器 [24, 30, 56, 60] 使用文本编码器对用户输入的在线词汇提示进行编码，并进行目标检测。值得注意的是，这些方法倾向于使用具有较重主干的大型检测器，例如 Swin-L [32]，以增加开放词汇容量。相比之下，“提示-检测”范式（图 2 (c)）首先对用户的提示进行编码，构建离线词汇表，该词汇表会根据不同的需求而变化。然后，高效的检测器可以动态推断离线词汇表，而无需重新编码提示。在实际应用中，一旦我们训练好了检测器（即 YOLO-World），就可以对提示或类别进行预编码，构建离线词汇表，然后将其无缝集成到检测器中。

我们的主要贡献可以概括为三个方面：

• 我们推出了 YOLO-World，这是一款先进的开放词汇目标检测器，在实际应用中非常高效。

• 我们提出了一个可重新参数化的视觉-语言 PAN，用于连接视觉和语言特征，并提出了一个开放词汇区域-文本对比的 YOLO-World 预训练方案。

• 我们提出的 YOLO-World 在大规模数据集上进行了预训练，展现了强大的零样本性能，在 LVIS 上实现了 35.4 的 AP 和 52.0 FPS 的帧率。预训练的 YOLO-World 可以轻松应用于下游任务，例如开放词汇实例分割和指称对象检测。此外，YOLO-World 的预训练权重和代码将开源，以促进更多实际应用。

2. 相关工作

2.1. 传统目标检测

目前流行的目标检测研究主要集中在固定词汇（闭集）检测上，即在预定义类别的数据集上训练目标检测器，例如 COCO 数据集 [26] 和 Objects365 数据集 [46]，然后检测固定类别集合内的目标。在过去的几十年中，传统目标检测方法可以简单地分为三类：基于区域的方法、基于像素的方法和基于查询的方法。基于区域的方法 [12, 13, 17, 27, 44]，例如 Faster R-CNN [44]，采用两阶段框架进行候选框生成 [44] 以及基于感兴趣区域 (RoI) 的分类和回归。基于像素的方法 [28, 31, 42, 49, 62] 往往是单阶段检测器，它们对预先定义的锚点或像素进行分类和回归。DETR [1] 首次探索了通过 Transformer [50] 进行目标检测，并启发了大量基于查询的方法 [65]。在推理速度方面，Redmon 等人提出了 YOLO [40–42]，利用简单的卷积架构进行实时目标检测。一些研究 [11, 23, 33, 52, 55] 提出了各种 YOLO 架构或设计，包括路径聚合网络 [29]、跨阶段部分网络 [51] 和重新参数化 [7]，这些方法进一步提高了速度和准确性。与之前的 YOLO 相比，本文中的 YOLOWorld 旨在检测固定词汇表之外的目标，并具有强大的泛化能力。

2.2.开放词汇目标检测

开放词汇目标检测 (OVD) [59] 已成为现代目标检测的新趋势，旨在检测超出预定义类别的目标。早期研究 [9, 14, 48, 53, 58] 遵循标准 OVD 设置 [59]，在基类上训练检测器，并评估新的（未知）类别。尽管如此，这种开放词汇设置可以评估检测器检测和识别新物体的能力，但它在开放场景下仍然受到限制，并且由于在有限的数据集和词汇上进行训练，缺乏对其他领域的泛化能力。Detic [64] 结合图像分类数据集 [5] 来扩展大词汇检测的对象类别。受到视觉语言预训练 [19, 39] 的启发，最近的研究 [35, 36, 56, 57, 60, 63] 将开放词汇对象检测公式化为图像文本匹配，并利用大规模图像文本数据来大规模增加训练词汇量。OWL-ViTs [35, 36] 使用检测和大规模自动标记数据集对简单的视觉转换器 [8] 进行微调，并构建简单的开放词汇检测器。GLIP [24] 提出了一种基于短语基础的开放词汇检测预训练框架，并在零样本设置下进行评估。接地 DINO [30] 将接地预训练 [24] 融入到具有跨模态融合的检测变压器 [61] 中。一些方法 [25, 56, 57, 60] 通过区域-文本匹配将检测数据集和图文数据集统一起来，并使用大规模图文对预训练检测器，取得了良好的性能和泛化能力。然而，这些方法通常使用像 ATSS [62] 或 DINO [61] 这样的大型检测器，并以 Swin-L [32] 为骨干，这导致了高计算需求和部署挑战。相比之下，我们提出了 YOLO-World，旨在实现高效的开放词汇目标检测，并实现实时推理，并简化下游应用的部署。与 ZSD-YOLO [54]（它也通过语言模型对齐来探索使用 YOLO 进行开放词汇检测 [59]）不同，YOLO-World 引入了一个新颖的 YOLO 框架，该框架具有有效的预训练策略，增强了开放词汇的性能和泛化能力。

3. 方法

3.1. 预训练公式：区域-文本对

传统的目标检测方法，包括 YOLO 系列 [20]，都是使用实例标注 Ω = {Bi, ci}N i=1 进行训练的，该标注由边界框 {Bi} 和类别标签 {ci} 组成。本文将实例标注重新表述为区域-文本对 Ω = {Bi, ti}N i=1，其中 ti 是区域 Bi 对应的文本。具体而言，文本 ti 可以是类别名称、名词短语或目标描述。YOLO-World 以图像 I 和文本 T 作为输入，输出预测框 {Bˆk} 和对应的目标嵌入 {ek} (ek ∈ RD)。

3.2.模型架构

YOLO-World 的整体架构如图 3 所示，它由 YOLO 检测器、文本编码器和可重新参数化的视觉语言路径聚合网络 (RepVL-PAN) 组成。给定输入文本，YOLO-World 中的文本编码器将文本编码为文本向量。YOLO 检测器中的图像编码器从输入图像中提取多尺度特征。然后，我们利用 RepVL-PAN，通过利用图像特征和文本向量之间的跨模态融合来增强文本和图像的表征。

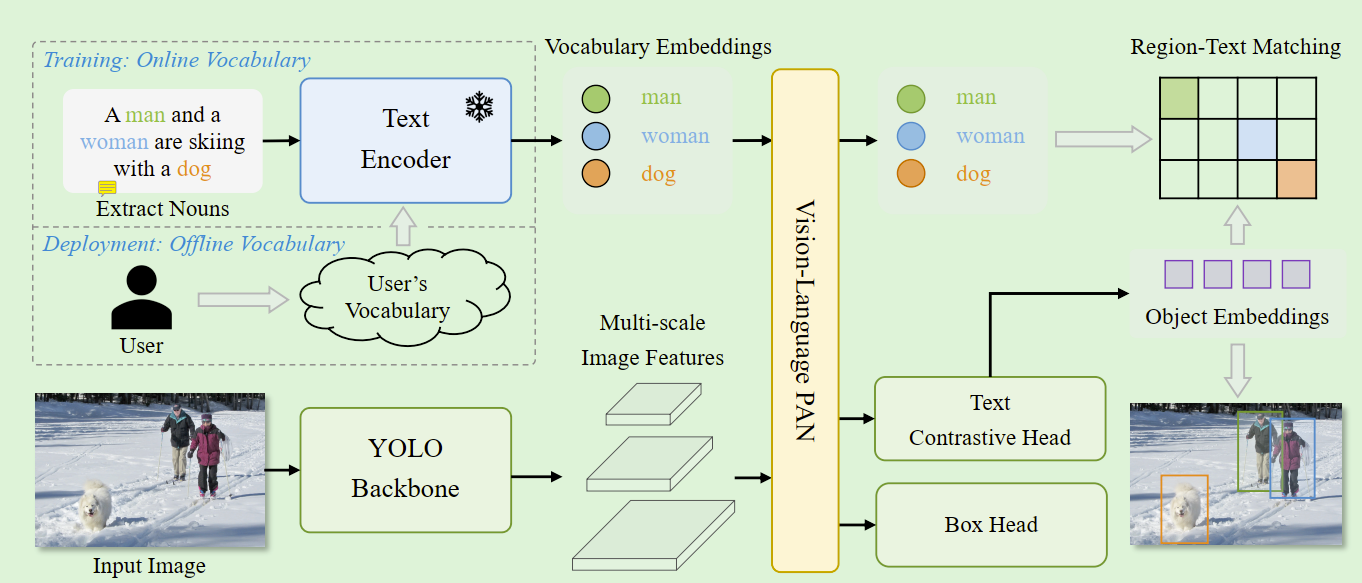


图 3. YOLO-World 的整体架构。与传统的 YOLO 检测器相比，YOLO-World 作为开放词汇检测器，采用文本作为输入。文本编码器首先对输入文本进行编码，生成文本向量。然后，图像编码器将输入图像编码为多尺度图像特征，而提出的 RepVL-PAN 利用多级跨模态融合技术对图像和文本特征进行融合。最后，YOLO-World 预测回归的边界框和对象嵌入，以匹配输入文本中出现的类别或名词。

YOLO 检测器。YOLO-World 主要基于 YOLOv8 [20] 开发，其中包含一个 Darknet 主干 [20, 43] 作为图像编码器，一个用于多尺度特征金字塔的路径聚合网络 (PAN)，以及一个用于边界框回归和对象嵌入的头部。文本编码器。给定文本 T，我们采用 CLIP [39] 预训练的文本编码器提取相应的文本向量 W = TextEncoder(T) ∈ RC×D，其中 C 表示名词数量，D 表示向量维度。与纯文本语言编码器 [6] 相比，CLIP 文本编码器在连接视觉对象和文本方面提供了更佳的视觉语义能力。当输入文本为标题时，我们采用简单的名词提取算法提取名词短语并将其输入文本编码器。

文本对比头。借鉴前人研究 [20]，我们采用包含两个 3×3 卷积层的解耦头，对边界框 {bk}K k=1 和对象向量 {ek}K k=1 进行回归，其中 K 表示对象数量。我们提出了一个文本对比头，通过以下公式获得对象-文本相似度 sk,j：



其中 L2-Norm(·) 是 L2 正则化，wj ∈ W 是第 j 个文本向量。此外，我们添加了具有可学习缩放因子 α 和平移因子 β 的仿射变换。L2 正则化和仿射变换对于区域-文本训练的稳定性都至关重要。

使用在线词汇进行训练。在训练过程中，我们为每个包含 4 幅图像的马赛克图像样本构建一个在线词汇表 T。具体来说，我们从马赛克图像中抽取所有正面名词，并从相应的数据集中随机抽取一些负面名词。每个马赛克图像样本的词汇表最多包含 M 个名词，M 默认值为 80。

使用离线词汇进行推理。在推理阶段，我们提出了一种使用离线词汇表的“提示-检测”策略，以提高效率。如图 3 所示，用户可以定义一系列自定义提示，其中可能包含标题或类别。然后，我们利用文本编码器对这些提示进行编码，并获取离线词汇嵌入。离线词汇可以避免每次输入的计算，并可以根据需要灵活地调整词汇。

3.3. 可重新参数化的视觉语言 PAN

图 4 展示了所提出的 RepVL-PAN 的结构，它遵循 [20, 29] 中自上而下和自下而上的路径，利用多尺度图像特征 {C3, C4, C5} 构建特征金字塔 {P3, P4, P5}。此外，我们提出了文本引导的 CSPLayer（T-CSPLayer）和图像池化注意力机制（I-Pooling Attention），以进一步增强图像特征和文本特征之间的交互，从而提升开放词汇的视觉语义表征能力。在推理过程中，离线词汇嵌入可以重新参数化为卷积层或线性层的权重以供部署。

文本引导的 CSPLayer。如图 4 所示，跨阶段局部层（CSPLayer）在自上而下或自下而上的融合之后使用。我们扩展了 [20] 中的 CSPLayer（也称为 C2f），将文本引导融入多尺度图像特征中，形成文本引导的 CSPLayer。具体而言，给定文本嵌入 W 和图像特征 Xl ∈ RH×W×D (l ∈ {3, 4, 5})，我们在最后一个暗瓶颈块之后采用最大 S 型注意力机制，将文本特征聚合到图像特征中，具体如下：

其中，更新后的 Xl′ 与跨阶段特征连接作为输出。δ 表示 S 型函数。

图像池化注意力机制。为了利用图像感知信息增强文本嵌入，我们提出了图像池化注意力机制，通过聚合图像特征来更新文本嵌入。我们并没有直接对图像特征使用交叉注意力机制，而是利用多尺度特征的最大池化来获取 3×3 区域，最终得到总共 27 个块标记 X˜ ∈ R27×D。然后，文本嵌入通过以下方式更新：

图 4. RepVL-PAN 示意图。我们提出的 RepVLPAN 采用文本引导 CSPLayer（T-CSPLayer）将语言信息注入图像特征，并采用图像池化注意力机制（I-Pooling Attention）来增强文本嵌入。

3.4. 预训练方案

本节将介绍在大规模检测、基础数据集和图文数据集上对 YOLO-World 进行预训练的训练方案。

从区域 - 文本对比损失中学习。给定马赛克样本 I 和文本 T，YOLO-World 输出 K 个对象预测 {Bk，sk}K k=1 以及注释 Ω = {Bi，ti}N i=1。我们遵循[20]并利用任务对齐的标签分配[10]将预测与地面实况注释相匹配，并为每个正预测分配一个文本索引作为分类标签。基于此词汇表，我们通过对象文本（区域 - 文本）相似性和对象文本分配之间的交叉熵构建了区域 - 文本对比损失 Lcon 。此外，我们采用 IoU 损失和分布式焦点损失进行边界框回归，总训练损失定义为：L（I）= Lcon + λI·（Liou + Ldfl），其中 λI 是指示因子，当输入图像 I 来自检测或接地数据时设置为 1，当它来自图像文本数据时设置为 0。考虑到图文数据集中存在噪声边界框，我们仅计算具有精确边界框的样本的回归损失。

基于图文数据的伪标注。我们并非直接使用图文对进行预训练，而是提出了一种自动标注方法来生成区域-文本对。具体而言，该标注方法包含三个步骤：(1) 提取名词短语：我们首先利用名词提取算法从文本中提取名词短语；(2) 伪标注：我们采用预先训练的开放词汇检测器，例如 GLIP [24]，为每幅图像中给定的名词短语生成伪边界框，从而提供粗略的区域-文本对。(3) 过滤：我们使用预先训练的 CLIP [39] 来评估图文对和区域-文本对的相关性，并过滤掉相关性较低的伪标注和图像。我们进一步结合非极大值抑制 (NMS) 等方法过滤掉冗余的边界框。我们建议读者参考附录了解详细方法。通过上述方法，我们从 CC3M [47] 中采样并标记了 246k 张图像，并使用 821k 个伪注释来构建 CC3M-Lite 数据集。

4. 实验

在本节中，我们将通过在大规模数据集上对提出的 YOLO-World 进行预训练，并在 LVIS 基准和 COCO 基准上以零样本方式评估 YOLO-World，以证明其有效性（参见 4.2 节）。我们还评估了 YOLOWorld 在 COCO 和 LVIS 数据集上进行目标检测的微调性能。

4.1. 实现细节

YOLO-World 基于 MMYOLO 工具箱 [3] 和 MMDetection 工具箱 [2] 开发。遵循 [20] 的指导，我们针对不同的延迟要求提供了三种 YOLO-World 变体，例如小型 (S)、中型 (M) 和大型 (L)。我们采用开源 CLIP [39] 文本编码器（带有预训练权重）对输入文本进行编码。除非另有说明，否则我们在单个 NVIDIA V100 GPU 上测量所有模型的推理速度，无需额外的加速机制（例如 FP16 或 TensorRT）。

4.2. 预训练

实验设置。在预训练阶段，我们采用 AdamW 优化器 [34]，初始学习率为 0.002，权重衰减为 0.05。YOLO-World 在 32 块 NVIDIA V100 GPU 上预训练了 100 个 epoch，总批次大小为 512。在预训练过程中，我们遵循先前的研究 [20]，采用颜色增强、随机仿射、随机翻转和 4 幅图像的马赛克算法进行数据增强。文本编码器在预训练期间处于冻结状态。

预训练数据。对于 YOLO-World 的预训练，我们主要采用检测数据集或基础数据集，包括 Objects365 (V1) [46]、GQA [18] 和 Flickr30k [38]，如表 1 所示。 1. 参照 [24]，我们在 GoldG [21]（GQA 和 Flickr30k）中排除了 COCO 数据集中的图像。此外，我们还使用图文对扩展了预训练数据，即 CC3M-Lite，我们从 CC3M [47] 中采样并标记了 246,000 张图像。

零样本评估。预训练后，我们以零样本方式在 LVIS 数据集 [15] 上评估了所提出的 YOLO-World。LVIS 数据集包含 1203 个目标类别，远多于预训练检测数据集的类别数，可以衡量其在大词汇量检测中的表现。参照前人的研究 [21, 24, 56, 57]，我们主要在 LVIS minival [21] 上进行评估，并报告了 Fixed AP [4] 的结果以供比较。最大预测数量设置为 1000。