全景分割变换器：深入探索基于Transformer模型的全景分割

李志奇[1], 王文海[2],谢恩泽[3],于志定[4],阿尼玛·阿南德库马尔[4,5],何塞·M·阿尔瓦雷斯[4],罗平[3],卢通[1]南京大学 [2]上海人工智能实验室 [3]香港大学 [4]英伟达 [5]加州理工学院 lzq@smail.nju.edu.cn wangwenhai@pjlab.org.cn xieenze@hku.hk zhidingy@nvidia.com aanandkumar@nvidia.com josea@nvidia.com pluo@cs.hku.hk lutong@nju.edu.cn

摘要

全景分割结合了联合语义分割与实例分割，将图像内容划分为可数物体和背景两类。我们提出全景分割变换器，这是一个基于*Transformer*模型的全景分割通用框架。它包含三个创新组件：高效的深度监督掩码解码器、查询解耦策略以及改进的后处理方法。我们还采用可变形*DETR*高效处理多尺度特征，这是*DETR*的快速高效版本。具体而言，我们以逐层方式监督掩码解码器中的注意力模块。这种深度监督策略使注意力模块能快速聚焦于有意义的语义区域，不仅提升了性能，相比可变形*DETR*还将所需训练轮次减少了一半。我们的查询解耦策略通过解耦查询集的职责，避免了可数物体与背景之间的相互干扰。此外，我们的后处理策略通过综合考虑分类与分割质量来解决掩码重叠冲突，在不增加额外成本的情况下提升了性能。该方法将基线*DETR*模型的全景质量提升了*6.2%*。全景分割变换器在*COCO*测试开发集上以*56.2%*的全景质量取得了最先进的结果，同时展现出比现有方法更强的零样本鲁棒性。

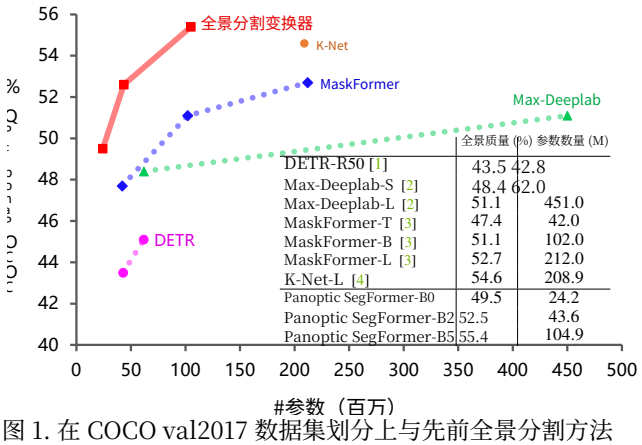

图 1. 在 COCO val2017 数据集划分上与先前全景分割方法的比较。Panoptic SegFormer 模型在不同模型中优于其他对应模型。Panoptic SegFormer (PVTv2-B5 [5]) 实现了 55.4% 的全景质量，以显著更少的参数量超越了先前的方法。

1. 引言

语义分割和实例分割是两个重要且相关的视觉任务。它们的内在联系最近催生了全景分割，作为这两项任务的统一 [6]。在全景分割中，图像内容被分为两种类型：可数物体和背景。可数物体指的是可数实例（例如，人、汽车），每个实例都有一个唯一的ID以区别于其他实例。背景指的是无定形且不可数的区域（例如，天空、草地），没有实例ID [6]。

近期研究 [1–3]尝试利用 Transformer模型 通过一个查询集来处理可数物体和背景。例如，DETR [1] 通过在一个端到端目标检测器之上添加一个全景分割头，简化了全景分割的工作流程。与先前的方法 [6, 7], 不同，DETR 不需要额外的手工设计的流程 [8, 9]。虽然简单，但 DETR 也带来了一些问题：(1) 它需要漫长的训练过程才能收敛；(2) 由于自注意力机制的计算复杂度与输入序列长度的平方成正比，DETR 的特征分辨率受到限制。因此它使用一个 FPN风格 [1, 10] 的全景分割头来生成掩码，这通常会导致边界保真度较低；(3) 它平等地处理可数物体和背景，但用边界框来表示它们，这对于背景 [2, 3] 可能不是最优的。尽管 DETR 在目标检测任务上取得了优异的性能，但其在全景分割上的优势尚未得到充分证明。为了

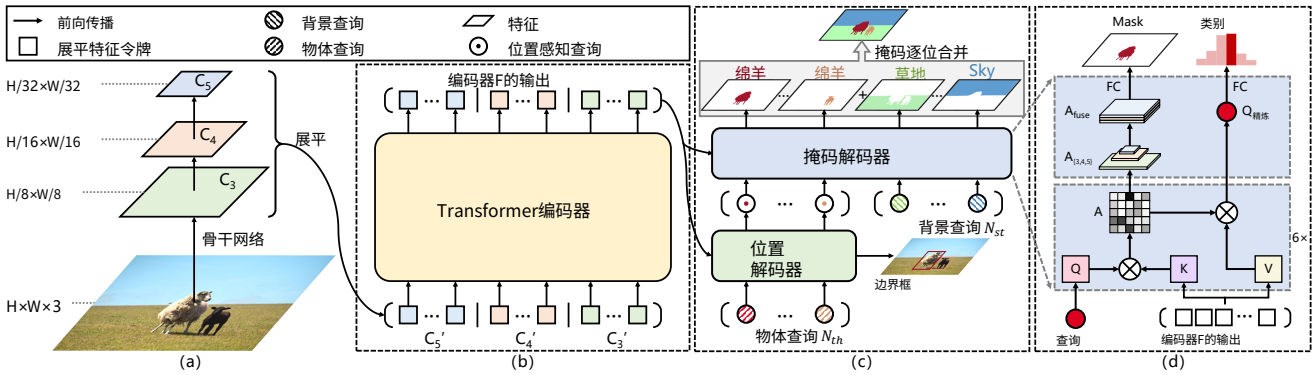

图 2. **Panoptic SegFormer 概述**。Panoptic SegFormer 由骨干网络、编码器和解码器组成。骨干网络和编码器输出并精炼多尺度特征。位置解码器的输入是 N_{th} 物体查询和多尺度特征。我们将来自位置解码器的 N_{th} 物体查询和 N_{st} 背景查询输入到掩码解码器。位置解码器的目标是学习查询的参考点，而掩码解码器则预测最终的类别和掩码。解码器的细节将在下文介绍。我们使用掩码级合并方法，而非常用的像素级argmax方法来进行推理。

克服 DETR 在全景分割上的缺陷，我们提出了一系列新颖有效的策略，这些策略大幅提升了基于Transformer模型 的全景分割模型的性能。

我们的方法。 在这项工作中，我们提出了全景分割变换器（Panoptic SegFormer），一个用于Transformer模型全景分割的简洁而有效的框架。我们的框架设计基于以下观察：1）深度监督对于在掩码解码器中学习高质量的判别性注意力表示至关重要。2）使用相同的方案处理可数物体和背景 [1] 是次优的，因为可数物体和背景之间存在不同的特性 [6]。3）常用的后处理方法，如像素级Argmax [1–3]，由于极端异常值，往往会产生假阳性结果。我们在全景分割变换器框架中通过以下方式克服这些挑战：

• 我们提出了一个掩码解码器，它利用多尺度注意力图来生成高保真掩码。该掩码解码器采用深度监督，在中间层促进判别性注意力表示，从而获得更好的掩码质量和更快收敛。

• 我们提出了一种查询解耦策略，将查询集分解为一个通过二分匹配来匹配可数物体的物体查询集，以及另一个通过类别固定分配来处理背景的背景查询集。该策略避免了每个查询内部可数物体和背景之间的相互干扰，并显著提高了背景分割的质量。更多细节请参阅第3.3.1节和图3。

• 我们提出了一种改进的后处理方法，用于生成全景格式的结果。除了比广泛使用的像素级Argmax方法更高效外，我们的方法还包含一种掩码级合并策略，该策略同时考虑了分类概率和预测掩码质量。仅我们的后处理方法就为DETR带来了1.3%的PQ改进。 [1].

我们在COCO数据集上进行了大量实验。如图 [11] 1所示，全景分割变换器以少得多的参数量显著超越了MaskFormer [3] 和K-Net等先前技术。借助可变形注意力 [12] 和我们深度监督的掩码解码器，我们的方法所需的训练轮次远少于之前的基于Transformer的方法（24轮对比. 300+轮）。此外，我们的方法在实例分割任务上也与当前方法取得了具有竞争力的性能。[13, 14] 。

2. 相关工作

全景分割。 全景分割已成为整体场景理解的热门任务 [6,15–17]。全景分割的相关研究主要将这一问题视为实例分割和语义分割的联合任务，其中可数物体和背景被分开处理 [18, 19]。Kirillov 等人。 [6] 提出了全景分割的概念和基准，并提供了一个基线方法，该方法直接结合了单独的实例分割模型和语义分割模型的输出。此后，诸如全景特征金字塔网络 [7], 、统一全景分割网络 [9]和注意力U型网络 [20] 等模型通过将实例分割和语义分割整合到单一模型中，提高了精度并减少了计算开销。然而，这些方法通过解决替代的子任务来逼近目标任务，因此引入了不必要的模型复杂性和次优性能。

最近，人们致力于统一全景分割的框架。Li 等人 [21] 提出了Panoptic FCN，其中全景分割流程通过一种类似于CondInst [22]的“自上而下与自下而上相遇”的双分支设计得以简化。在他们的工作中，可数物体和背景通过一个对象/区域级内核分支和一个图像级特征分支联合建模。最近的几项工作将可数物体和背景表示为查询，并通过Transformer模型执行端到端全景分割。

DETR [1]预测可数物体和背景的边界框，并组合Transformer解码器的注意力图和ResNet [23] 的特征图来执行全景分割。Max-Deeplab [2]通过一个双路径Transformer直接预测对象类别和掩码，无论类别是可数物体还是背景。在DETR的基础上，MaskFormer [3] 使用了一个额外的像素解码器来精炼高空间分辨率的特征，并通过将查询与像素解码器的特征相乘来生成掩码。由于自注意力 [24],的计算复杂度，DETR和MaskFormer都使用空间分辨率有限的特征图进行全景分割，这损害了性能，并需要在最终的掩码预测中结合额外的高分辨率特征图。与上述方法不同，我们的查询解耦策略使用独立的查询集处理可数物体和背景。尽管可数物体查询和背景查询是为不同目标设计的，但它们由掩码解码器以相同的工作流程处理。这些查询的预测结果格式相同，因此我们可以在后处理过程中以平等的方式处理它们。一项同期工作 [4]采用了类似思路，使用动态内核执行实例和语义分割，旨在利用统一的内核处理各种分割任务。与之相反，我们的目标是更深入地研究基于Transformer的全景分割。由于各种任务性质不同，统一的流程是否适合这些任务仍然是一个开放性问题。在本工作中，我们利用一个额外的位置解码器来辅助可数物体学习位置线索并获得更好的结果。

端到端目标检测。近期流行的端到端目标检测框架启发了许多其他相关研究 [13, 25]。其中，DETR [1] 可以说是这些方法中最具代表性的端到端目标检测器。DETR将目标检测任务建模为一个带有可学习查询的字典查找问题，并采用编码器-解码器Transformer来预测边界框，无需额外的后处理。DETR极大地简化了传统的检测框架，并移除了许多手工设计的组件，例如非极大值抑制 (NMS) [26, 27] 和锚框 [27]。Zhu 等人. [12] 提出了可变形DETR，它通过可变形注意力层进一步降低了内存和计算成本。在本工作中，我们采用可变形注意力 [12]，因为它相比DETR [1]具有更高的效率和更快的收敛速度。

3. 方法

3.1. 整体架构

如图2所示，全景分割变换器由三个关键模块组成：Transformer编码器、位置解码器和掩码解码器，其中（1）Transformer编码器用于精炼骨干网络提供的多尺度特征图，

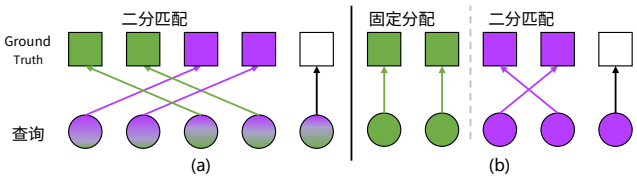

图3. (a) 方法 [1–3] 采用一个查询集来联合匹配可数物体（紫色方块）和背景（绿色方块）。(b) 我们使用一个物体查询集（紫色圆圈）通过二分匹配来定位可数物体，并使用一个背景查询集（绿色圆圈）通过类别固定分配策略来预测背景。 ∅ 被分配给未匹配的查询。

（2）位置解码器旨在捕获可数物体的位置线索，（3）掩码解码器用于最终的分类和分割。

我们的架构将输入图像 $X \in \mathbb{R}^{H \times W \times 3}$ 馈送到骨干网络，并从最后三个阶段获取特征图 C_3、C_4 和 C_5，其分辨率相对于输入图像分别为 $1/8$、$1/16$ 和 $1/32$。我们通过一个全连接层将这三个特征图投影到具有 256 个通道的特征图，并将它们展平为特征令牌 C'_3、C'_4 和 C'_5。这里，我们将 L_i 定义为 $\frac{H}{2^{i+2}} \times \frac{W}{2^{i+2}}$，而 C'_3、C'_4 和 C'_5 的形状分别为 $L_1 \times 256$、$L_2 \times 256$ 和 $L_3 \times 256$。接下来，使用拼接后的特征令牌作为输入，变换器编码器输出尺寸为 $(L_1+L_2+L_3)\times 256$ 的精炼特征。之后，我们分别使用 N_{th} 个和 N_{st} 个随机初始化的物体查询和背景查询来描述可数物体和背景。位置解码器通过检测物体的边界框来精炼 N_{th} 个物体查询，以捕获位置信息。随后，掩码解码器将物体查询和背景查询同时作为输入，并在每一层预测掩码和类别。

在推理过程中，我们采用掩码级合并策略，将最终掩码解码器层预测的掩码转换为全景分割结果，具体细节将在第3.5节详细介绍。

3.2. Transformer编码器

高分辨率和多尺度特征图对于分割任务至关重要 [7, 21, 28]。由于自注意力层的高计算成本，以往的基于Transformer的方法 [1, 3] 只能在其编码器中处理低分辨率特征图（例如，ResNet C_5），这限制了分割性能。与这些方法不同，我们采用可变形注意力 [12] 来实现我们的Transformer编码器。得益于可变形注意力的低计算复杂度，我们的编码器能够精炼高分辨率和多尺度特征图并为其引入位置编码 [24] 。 F

3.3. 解码器

在本节中，我们首先介绍我们的查询解耦策略，然后详细解释我们的位置解码器和掩码解码器。

3.3.1 查询解耦策略

我们认为，使用一个查询集来同等处理可数物体和背景是次优的。由于它们之间存在许多不同的属性，可数物体和背景很可能相互干扰并损害模型性能，尤其是对于全景质量$^{\text{st}}$。为了防止可数物体和背景相互干扰，我们在全景分割变换器中应用了查询解耦策略，如图3所示。具体来说，N_{th}物体查询用于预测可数物体的结果，而N_{st}背景查询仅针对背景。采用这种形式，物体查询和背景查询可以共享相同的处理流程，因为它们的格式相同。我们也可以根据不同任务的特点，为可数物体或背景定制私有的工作流程。在本工作中，我们使用一个额外的位置解码器来检测单个实例（使用物体查询），这将有助于区分不同的实例 [6]。掩码解码器同时接受物体查询和背景查询，并生成最终的掩码和类别。请注意，对于物体查询，真实标注通过二分图匹配策略进行分配。对于背景，我们使用类别固定分配策略，每个背景查询对应一个背景类别。物体查询和背景查询将以相同的格式输出结果，我们使用统一的后处理方法处理这些结果。

3.3.2 位置解码器

在全景分割任务中，位置信息在区分具有不同实例ID的可数物体方面起着重要作用 [22, 28, 29]。受此启发，我们采用一个位置解码器，将可数物体的位置信息引入到可学习的查询中。具体来说，给定 N_{th}随机初始化的物体查询和由Transformer编码器生成的精炼特征令牌，该解码器将输出 N_{th} 位置感知查询。

在训练阶段，我们在位置感知查询之上应用一个辅助的多层感知机头，以预测目标对象的边界框和类别。我们使用检测损失 $\mathcal{L}_{\text{det}}$来监督预测结果。该多层感知机头是一个辅助分支，可以在推理阶段丢弃。位置解码器遵循可变形DETR [12]。值得注意的是，位置解码器可以通过预测掩码的质心而不是边界框来学习位置信息。这种无框模型仍然可以达到与我们基于框的模型相当的结果。

3.3.3 掩码解码器

如图2 (d)所示，掩码解码器被提出来根据给定的查询预测类别和掩码。掩码解码器的查询 Q 是来自位置解码器的位置感知物体查询或类别固定的背景查询。掩码解码器的键向量 K 和值向量 V 是从Transformer编码器精炼后的特征令牌 F投影得到的。我们首先将物体查询通过掩码解码器，然后获取注意力图 $A \in \mathbb{R}^{N\times h\times(L_1+L_2+L_3)}$ 以及来自每个解码器层的精炼查询 $Q_{\text{refine}} \in \mathbb{R}^{N\times 256}$，其中 $N = N_{\text{th}} + N_{\text{st}}$ 是查询总数，h 是注意力头数量，$L_1+L_2+L_3$ 是特征令牌 F的长度。

与 [1, 2], 等方法类似，我们直接在每个解码器层精炼后的查询 Q_{refine}之上通过一个全连接层进行分类。每个物体查询需要预测所有物体类别的概率。背景查询仅预测其对应背景类别的概率。

同时，为了预测掩码，我们首先将注意力图 A 分割并重塑为注意力图 A_3、A_4 和 A_5，它们分别与 C_3、C_4 和 C_5 具有相同的空间分辨率。此过程可表述为：

$$(A_3, A_4, A_5) = \text{Split}(A), \quad A_i \in \mathbb{R}^{\frac{H}{2^{i+2}}\times\frac{W}{2^{i+2}}\times h}, \tag{1}$$

其中 Split(·) 表示分割和重塑操作。之后，如公式(2)所示，我们将这些注意力图上采样到 $H/8\times W/8$ 的分辨率，并沿通道维度进行拼接，

$$A_{\text{fused}} = \text{Concat}(A_1, \text{Up}_{\times 2}(A_2), \text{Up}_{\times 4}(A_3)), \tag{2}$$

其中 $\text{Up}_{\times 2}(\cdot)$ 和 $\text{Up}_{\times 4}(\cdot)$ 分别表示 2 倍和 4 倍的双线性插值操作。Concat(·) 是拼接操作。最后，基于融合后的注意力图 A_{fused}，我们通过一个 1×1 卷积来预测二值掩码。

先前的研究 [12] 认为，DETR收敛缓慢的原因在于注意力模块会平等地关注特征图中的所有像素，而学习聚焦于稀疏的有意义位置需要大量努力。我们在掩码解码器中采用两个关键设计来解决此问题：(1) 使用一个超轻量的全连接头从注意力图生成掩码，确保注意力模块能够被真实掩码引导以学习应聚焦于何处。该全连接头仅包含200 个参数量，这保证了注意力图的语义信息与掩码高度相关。直观地说，真实掩码正是我们希望注意力模块聚焦的有意义区域。(2) 我们在掩码解码器中采用深度监督。每一层的注意力图都将受到掩码的监督，使得注意力模块能够在更早的阶段捕获有意义的信息。这可以极大地加速注意力模块的学习过程。

3.4. 损失函数

在训练期间，我们的全景分割变换器的总体损失函数可以写为：

$$\mathcal{L} = \lambda_{\text{things}}\mathcal{L}_{\text{things}} + \lambda_{\text{stuff}}\mathcal{L}_{\text{stuff}}, \tag{3}$$

其中 L_{things} 和 L_{stuff} 分别是针对可数物体和背景的损失。λ_{things} 和 λ_{stuff} 是超参数。

物体损失。遵循常见做法 [1, 30], ，我们在预测集与真实标注集之间搜索最佳的二分匹配。

具体而言，我们利用匈牙利算法 [31] 来搜索具有最小匹配成本的排列，该成本是分类损失 $\mathcal{L}_{\text{cls}}$、检测损失 $\mathcal{L}_{\text{det}}$ 与分割损失$\mathcal{L}_{\text{seg}}$的总和。因此，物体类别的总体损失函数定义如下：

$$\mathcal{L}_{\text{things}} = \lambda_{\text{det}}\mathcal{L}_{\text{det}} + \sum_{i}^{D_m}(\lambda_{\text{cls}}\mathcal{L}^i_{\text{cls}} + \lambda_{\text{seg}}\mathcal{L}^i_{\text{seg}}), \quad (4)$$

其中 λ_{cls}、 λ_{seg} 和 λ_{loc} 是用于平衡三种损失的权重。 D_m 是掩码解码器中的层数。$\mathcal{L}^i_{\text{cls}}$ 是由 Focal 损失 [27], 实现的分类损失，而 $\mathcal{L}^i_{\text{seg}}$ 是由 Dice 损失 [32] 实现的分割损失。 $\mathcal{L}_{\text{det}}$ 是用于执行检测的可变形 DETR 的损失。

背景损失。我们对背景使用固定的匹配策略。因此，背景查询与背景类别之间存在一一对应的映射关系。背景类别的损失定义类似如下：

$$\mathcal{L}_{\text{stuff}} = \sum_{i}^{D_m}(\lambda_{\text{cls}}\mathcal{L}^i_{\text{cls}} + \lambda_{\text{seg}}\mathcal{L}^i_{\text{seg}}), \quad (5)$$

其中 $\mathcal{L}^i_{\text{cls}}$ 和 $\mathcal{L}^i_{\text{seg}}$ 与公式 (4) 中的相同。

3.5. 掩码级合并推理

全景分割要求为每个像素分配一个类别标签（或空值）和实例 ID（对于背景则忽略） [6]。全景分割的一个挑战在于它要求生成非重叠的结果。最近的方法 [1–3] 直接使用像素级 argmax 来确定每个像素的归属，这可以自然地解决重叠问题。尽管像素级 argmax 策略简单有效，但我们观察到，由于异常的像素值，它总是会产生假阳性结果。

与像素级Argmax在每个像素上解决冲突不同，我们提出掩码级合并策略，通过重新解决预测掩码之间的冲突。具体而言，我们使用掩码的置信度分数来确定重叠区域的归属。受先前NMS方法 [28], 的启发，置信度分数同时考虑了分类概率和预测掩码质量。第i个结果的置信度分数可表述为：

$$s_i = p_i^{\alpha} \times \text{average}(\mathbb{1}_{\{m_i[h,w]>0.5\}} m_i[h,w])^{\beta}, \quad (6)$$

其中 p_i 是第i个结果最可能的类别概率。$m_i[h,w]$ 是像素 $[h,w]$处的掩码逻辑值， α, β用于平衡分类概率和分割质量的权重。

如算法1所示，掩码级合并策略以 c、 s和 m 作为输入，分别表示预测的类别、置信度分数和分割掩码。它输出一个语义掩码SemMsk 和一个实例ID掩码 IdMsk，为每个像素分配一个类别标签和一个实例ID。具体来说，SemMsk 和IdMsk 首先

算法1：掩码级合并

```
def MaskWiseMergeing(c,s,m):
    # 类别 c ∈ ℝ^N
    # 置信度分数 s ∈ ℝ^N
    # 掩码 m ∈ ℝ^{N×H×W}
    语义掩码 = np.zeros(H,W)
    ID掩码 = np.zeros(H,W)
    顺序 = np.argsort(-s)
    id = 0
    对于 i 按顺序:
        m_i =m[i]>0.5 & (语义掩码==0)
        如果s[i]<t_cnf 或 m_i / m[i]>0.5 < t_keep:
            继续
        语义掩码[m_i] = c[i]
        ID掩码[m_i] = id
        编号 += 1
返回语义掩码 ，ID掩码
```

初始化为零。然后，我们按置信度分数降序对预测结果进行排序，并按顺序将排序后的预测掩码填充到SemMsk和IdMsk中。接着，我们丢弃置信度分数低于t_{cls} 的结果，并移除与较低置信度分数重叠的部分。只有保留的、与原始掩码有足够比例t_{keep} 的非重叠部分才会被保留。最后，添加每个掩码的类别标签和唯一ID，以生成无重叠的全景格式结果。

4. 实验

我们在COCO数据集和ADE20K数据集上评估了全景分割变换器，并将其与多种先进方法进行比较。我们提供了全景分割和实例分割的主要结果，同时进行了详细的消融研究以验证各模块的效果。具体实现细节请参阅附录。

4.1. 数据集

我们在COCO 2017数据集上进行了实验 [11]，未使用外部数据。COCO数据集包含118K张训练图像和5K张验证图像，其中包含80类可数物体和53类背景。我们进一步在ADE20K数据集上验证了模型的通用性 [33], ，该数据集包含100类可数物体和50类背景。

4.2. 主要结果

全景分割。 我们在COCO数据集验证集和测试开发集上进行实验。在表1和表2中，我们报告了主要结果，并与其他最先进的方法进行了比较。全景分割变换器以ResNet-50作为骨干网络和单尺度输入，在COCO数据集验证集上取得了49.6%的全景质量，并且分别超越了先前的方法K-Net [4]和DETR [1] 超过2.5%和6.2%的全景质量。除了卓越的性能外，

方法	骨干网络	训练轮次	全景质量 PQ	PQ^{th}	PQ^{st}	#P	#F
全景特征金字塔网络 [7]	R50	36	41.5	48.5	31.1	-	-
SOLOv2 [28]	R50	36	42.1	49.6	30.7	-	-
DETR [1]	R50	325	43.4	48.2	36.3	42.9	248
全景FCN [21]	R50	36	43.6	49.3	35.0	37.0	244
K-Net [4]	R50	36	47.1	51.7	40.3	-	-
MaskFormer [3]	R50	300	46.5	51.0	39.8	45.0	181
全景分割变换器	R50	**12**	48.0	52.3	41.5	51.0	214
全景分割变换器	R50	24	**49.6**	**54.4**	**42.4**	51.0	214
DETR [1]	R101	325	45.1	50.5	37.0	61.8	306
Max-Deeplab-S [2]	Max-S [2]	54	48.4	53.0	41.5	61.9	162
MaskFormer [3]	R101	300	47.6	52.5	40.3	64.0	248
全景分割变换器	R101	24	**50.6**	**55.5**	**43.2**	69.9	286
Max-Deeplab-L [2]	Max-L [2]	54	51.1	57.0	42.2	451.0	1846
全景FCN [36]	Swin-L†	36	51.8	58.6	41.6	-	-
MaskFormer [3]	Swin-L†	300	52.7	58.5	44.0	212.0	792
K-Net [4]	Swin-L†	36	54.6	60.2	46.0	208.9	-
全景分割变换器	Swin-L†	24	**55.8**	**61.7**	**46.9**	221.4	816
全景分割变换器	PVTv2-B5†	24	55.4	61.2	46.6	104.9	349

表 1. 在 COCO 验证集上的实验。#P 和 #F 分别表示参数量 (M) 和浮点运算次数 (G)。全景分割变换器 (R50) 在 COCO 验证集上取得了 49.6% 的全景质量，超越了之前的方法，如 DETR [1] 和 MaskFormer [3]，分别高出 6.2% 全景质量和 3.1% 全景质量。† 表示骨干网络是在 ImageNet-22K 上预训练的。

方法	骨干网络	PQ	PQ^{th}	PQ^{st}	SQ	RQ
Max-Deeplab-L [2]	Max-L [2]	51.3	57.2	42.4	82.5	61.3
创新方法 [35]	集成	53.5	61.8	41.1	**83.4**	63.3
MaskFormer [3]	Swin-L†	53.3	59.1	44.5	82.0	64.1
K-Net [4]	Swin-L†	55.2	61.2	46.2	82.4	66.1
全景分割变换器	R50	50.2	55.3	42.4	81.9	60.4
全景分割变换器	R101	50.9	56.2	43.0	82.0	61.2
全景分割变换器	Swin-L†	**56.2**	**62.3**	**47.0**	82.8	**67.1**
全景分割变换器	PVTv2-B5†	55.8	61.9	46.5	83.0	66.5

表 2. 在 COCO测试开发集 上的实验。† 表示骨干网络在 ImageNet-22K 上进行了预训练。

方法	骨干网络	PQ	全景质量th	全景质量st	SQ	RQ
BGRNet [37]	R50	31.8	-	-	-	-
自动全景分割 [38]	ShuffleNetV2 [39]	32.4	-	-	-	-
MaskFormer [3]	R50	34.7	32.2	**39.7**	76.7	42.8
MaskFormer [3]	R101	35.7	34.5	38.0	77.4	43.8
全景分割变换器	R50	**36.4**	**35.3**	38.6	**78.0**	**44.9**

Ta表 3. ADE20K数据集验证集上的全景分割结果 set.

全景分割变换器是高效的。在 1× 训练策略（12个训练轮次）下，全景分割变换器（R50）取得了48.0%的全景质量，这比训练了300个训练轮次的MaskFormer [3]高出1.5%的全景质量。通过视觉变换器骨干网络Swin-L增强后 [34], , 全景分割变换器在COCO数据集测试开发集上取得了56.2%的全景质量的新记录，无需任何额外技巧，超越了MaskFormer [3] 超过2.9%的全景质量。我们的方法甚至超越了先前的竞赛级方法Innovation [35]超过2.7%的全景质量。通过采用PVTv2-B5 [5], , 我们也获得了可比的性能，同时与Swin-L相比，模型参数量和浮点运算次数显著减少。全景分割变换器在ADE20K数据集上也比MaskFormer高出1.7%的全景质量 [33], , 见表3。

实例分割。 全景分割变换器只需丢弃背景查询即可转换为实例分割模型。

方法	骨干网络	平均精度^{seg}	平均精度^{seg} S	平均精度^{seg} M	平均精度^{seg} L
掩码区域卷积神经网络 [40]	R50	37.5	21.1	39.6	48.3
SOLOv2 [28]	R50	38.8	16.5	41.7	56.2
K-Net [4]	R50	38.6	19.1	42.0	**57.7**
SOLQ [25]	R50	39.7	21.5	42.5	53.1
HTC [14]	R50	39.7	22.6	42.2	50.6
QueryInst [13]	R50	40.6	**23.4**	42.5	52.8
我们的方法（无裁剪）	R50	40.4	21.1	43.8	54.7
我们的方法（带裁剪）	R50	**41.7**	21.9	**45.3**	56.3

表4. COCO测试开发集上的实例分割。

	训练轮次	PQ	参数量	浮点运算次数	FPS
基线 (DETR [1])	325	43.4	42.9M	247.5G	4.9
+掩码级合并	325	44.7	42.9M	247.5G	6.1
++多尺度可变形注意力 [12]	50	47.3	44.9M	618.7G	2.7
+++掩码解码器	24	48.5	51.0M	214.8G	7.8
++++查询解耦	24	49.6	51.0M	214.2G	7.8

表 5. 我们将 DETR 的全景分割性能 [1] (R50 [23]) 从 43.4% PQ 提升至 49.6% PQ，同时训练轮次更少、计算成本更低、推理速度更快。

在表4中，我们报告了在COCO数据集`test-dev` 集上的实例分割结果。我们取得了与当前最先进的方法（如QueryInst [13] 和HTC [14], ）相当的结果，并且比K-Net [4]高出1.8 AP。在训练期间使用随机裁剪将AP提升了1.3个百分点。

4.3. 消融研究

首先，我们在表5中展示每个模块的效果。与基线 DETR 相比，我们的模型实现了更好的性能、更快的推理速度，并显著减少了训练轮次。我们默认使用全景分割变换器 (R50) 进行消融实验。

位置解码器的影响。位置解码器协助查询捕获可数物体的位置信息。表6展示了改变位置解码器中层数的结果。

#层	PQ	PQ^{th}	PQ^{st}
0	47.0	50.0	42.5
1	47.7	51.1	42.5
2	48.1	51.8	42.5
6* (无框)	49.2	53.5	42.6
6	49.6	54.4	42.4

表 6. 消融位置解码器。

当位置解码器层数较少时，我们的模型在可数物体上表现更差，这表明通过位置解码器学习位置线索有助于模型更好地处理可数物体。* 注：我们在位置解码器中预测的是质心而非边界框，这种无框模型取得了可比的结果（49.2% PQ对比 49.6% PQ）。

掩码级合并。 如表格 7所示，我们在多种模型上将我们的掩码级合并策略与像素级argmax策略进行了比较。我们同时使用掩码全景质量和边界全景质量 [41] 以使我们的结论更具可信度。采用掩码级合并策略的模型始终表现更优。采用掩码级合并的DETR比原始DETR在PQ上高出1.3% [1]。此外，我们的掩码级合并比DETR的像素级argmax耗时少20%，因为DETR在其代码中使用了更多技巧，例如合并具有相同类别的背景以及迭代地移除

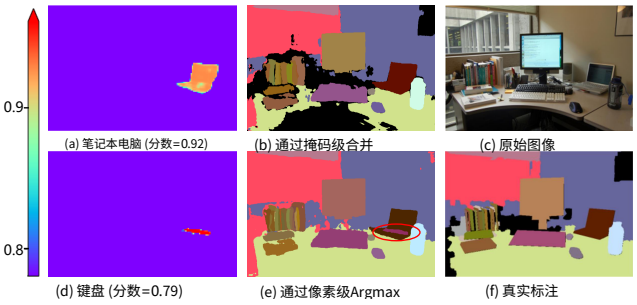

图 4. 在使用像素级argmax时，键盘被覆盖在笔记本电脑上（如(e)中红圈所示）。然而，笔记本电脑的分类概率高于键盘。像素级argmax策略未能利用这一重要线索。掩码逻辑值是通过 DETR-R50 [1]生成的。

方法	掩码全景质量 PQ	SQ	RQ	边界全景质量 [41] PQ	SQ	RQ
DETR (p)	43.4	79.3	53.8	32.8	71.0	45.2
DETR (m)	44.7	80.2	54.7	33.7	71.1	46.5
D-DETR-MS (p)	46.3	80.0	56.5	37.1	72.1	50.2
D-DETR-MS (m)	47.3	81.1	56.8	38.0	72.3	51.0
MaskFormer (p)	45.6	80.2	55.8	-	-	-
MaskFormer (p*)	46.5	80.4	56.8	36.8	72.5	49.8
MaskFormer (m)	46.8	80.4	57.2	37.6	72.6	51.1
全景分割变换器 (p)	48.4	80.7	58.9	39.3	73.0	52.9
全景分割变换器 (m)	49.6	81.6	59.9	40.4	73.4	54.2

表 7. 掩码级合并策略的效果。该表展示了采用不同后处理方法的模型结果，骨干网络为 ResNet-50。“(p)”指使用像素级argmax作为后处理方法。“(p*)”在其像素级argmax策略 [3]中同时考虑了类别概率和掩码预测概率。采用掩码级合并的“(m)”模型在掩码全景质量和边界全景质量 [41] 上始终优于像素级argmax方法。

方法	PQ	PQ^{th}	PQ^{st}	平均精度^{box}	平均精度^{seg}
DETR [1]	43.4	48.2	36.3	38.8	31.1
D-DETR-MS [12]	47.3	52.6	39.0	45.3	37.6
Panoptic FCN [21]	43.6	49.3	35.0	36.6	34.5
我们的方法（联合匹配）	48.5	**54.5**	39.5	44.1	37.7
我们的方法（查询解耦）	**49.6**	54.4	**42.4**	**45.6**	**39.5**

表8. 查询解耦策略的效果。各种全景分割模型在COCO val2017上的全景质量和平均精度分数。

面积较小的掩码。图 4 展示了一个使用像素级argmax策略的典型失败案例。

掩码解码器。我们提出的掩码解码器收敛更快，因为真实掩码引导注意力模块关注有意义的区域。图 5 展示了几个模型的收敛曲线。我们仅监督掩码解码器的最后一层，而未采用深度监督。我们可以观察到，我们的方法在训练24个训练轮次后达到49.6%的全景质量，更长的训练效果甚微。然而，D-DETR-MS至少需要50个训练轮次才能达到更好的性能。深度监督对于我们的掩码解码器获得更好性能和更快收敛至关重要。图 6 展示了

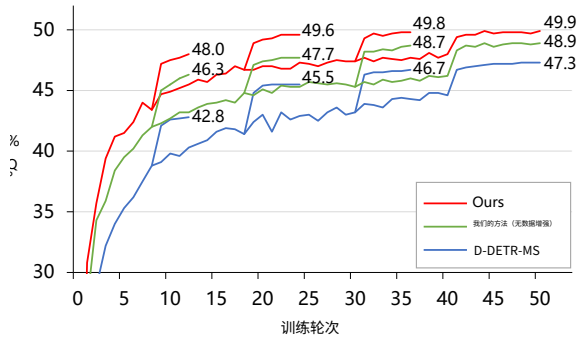

图5. Panoptic SegFormer 与 D-DETR-MS 的收敛曲线。我们使用不同的训练计划训练模型。“w/o ds”表示我们在掩码解码器中未采用深度监督。学习率在曲线跃升处降低。

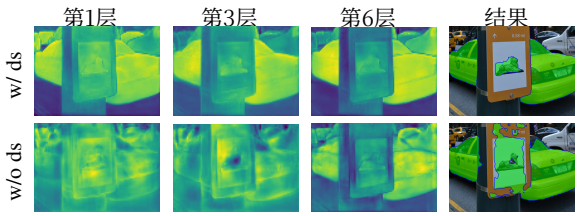

图6. 掩码解码器中不同层的注意力图。“ds”指深度监督。

掩码解码器中不同层的注意力图，当使用深度监督时，注意力模块会关注前一层中的目标汽车。这些注意力图与最终预测的掩码非常相似，因为掩码是由注意力图通过一个轻量级的全连接头生成的。

由于我们的掩码解码器可以从每一层生成掩码，我们评估了掩码解码器中每一层的性能，参见表10。在推理过程中，使用掩码解码器的前两层将与整个掩码解码器性能相当。由于计算成本降低，其推理速度也更快。全景质量th 几乎不受层数影响，全景质量st 在第一层表现稍差。原因是位置解码器对物体查询进行了额外的精炼。

层	PQ	PQ^{th}	PQ^{st}	Fps
1st	48.8	54.3	40.5	10.6
2nd	49.5	54.5	42.0	9.8
3rd	49.6	54.5	42.3	9.3
Last	49.6	54.4	42.4	7.8

表10. 掩码解码器中各层的结果。

查询解耦策略的效果。我们将我们提出的查询解耦策略与先前DETR的匹配方法（此处描述为“联合匹配”）进行比较 [1],，如表格8所示。遵循DETR，联合匹配使用一组查询来同时定位可数物体和背景，并将所有查询馈送到位置解码器和掩码解码器。对于我们提出的查询解耦策略，我们使用物体查询通过二分匹配检测可数物体，并使用位置解码器对它们进行精炼。背景查询通过类别固定分配策略进行分配。为了公平比较，联合匹配策略和我们的查询解耦策略都使用了353个查询。我们可以观察到

方法	清洁均值		Blur				噪声				数字				天气	
			运动	散焦	玻璃	高斯	脉冲	散粒	斑点	亮度	对比度	饱和度	JPEG	雪	斑点	雾	霜冻
全景FCN (R50)	43.8	26.8	22.5	23.7	14.1	25.0	28.2	20.0	28.3	31.9	39.4	24.3	38.0	22.9	20.0	29.6 35.3 25.3
MaskFormer (R50)	47.0	29.5	24.9	28.1	16.4	29.5	31.2	24.7	30.9	34.8	42.5	27.5	41.2	22.0	20.4	31.0 38.5 27.7
D-DETR (R50)	47.6	30.3	25.6	28.7	16.8	29.7	32.5	24.9	31.4	35.9	43.1	28.6	41.3	24.5	21.7	31.7 39.7 28.7
我们的方法 (R50)	50.0	32.9	26.9	30.2	17.5	31.6	35.5	27.9	35.4	38.6	45.7	31.3	43.9	29.0	24.3	35.0 41.9 32.3
MaskFormer (Swin-L)	52.9	41.7	37.3	38.0	30.4	39.3	42.3	42.5	42.8	45.3	49.7	43.9	49.4	39.7	35.2	45.2 48.8 37.9
我们的方法 (Swin-L)	**55.8**	**47.2**	41.3	41.5	34.3	42.7	48.6	49.5	48.8	50.3	53.8	50.1	53.5	46.9	44.8	51.5 53.3 44.3
我们的方法 (PVTv2-B5)	55.6	47.0	41.5	41.1	36.1	42.5	48.4	49.6	48.4	50.4	53.5	50.8	53.0	46.2	42.4	50.3 52.9 44.3

表 11. COCO-C 上的全景分割结果。为减轻实验工作量，我们使用了 COCO val2017 中的 2000 张图像子集。第三列是 16 种损坏数据上的平均结果。

我们提出的策略极大地提升了全景质量st。此外，全景分割模型可以仅利用其物体类别结果来执行实例分割。然而，先前的全景分割方法在实例分割任务上总是表现不佳，尽管这两个任务密切相关。表格8 展示了各种方法在全景分割和实例分割上的性能。我们的查询解耦策略可以在全景分割任务上实现最先进性能，同时获得具有竞争力的实例分割性能。

简而言之，与联合匹配相比，查询解耦策略实现了更高的全景质量st和平均精度^{seg} 。我们分析了联合匹配的实验结果，发现如果一个查询更偏好可数物体，那么它检测出的背景类别精度就会更低，参见

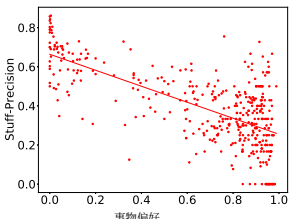

图 7. 物体类别偏好 *vs*. 背景类别精度。

图 7。每个点代表每个查询对应的事物偏好和背景类别精度，具体定义见附录。红线是这些点的线性回归。当使用一个查询集同时检测可数物体和背景时，会导致每个查询内部产生干扰。我们的查询解耦策略防止了可数物体和背景在同一查询内相互干扰。

4.4. 对自然损坏的鲁棒性

全景分割在自动驾驶等许多领域具有广阔的应用前景。模型鲁棒性是自动驾驶最关注的问题之一。在本实验中，我们评估了模型对扰动数据的鲁棒性。我们遵循 [42] 并生成了COCO-C，它扩展了COCO验证集，包含了由模糊、噪声、数字和天气四大类别的16种算法生成的扰动数据。我们将我们的模型与Panoptic FCN [21], 、D-DETR-MS和MaskFormer [3]进行比较。结果如表11所示。我们计算了COCO-C上扰动数据的平均结果。使用相同的骨干网络，我们的模型始终优于其他模型。先前的研究 [43–45]发现，基于Transformer的模型在图像分类和语义分割任务上具有更强的鲁棒性。

我们的实验结果也表明，基于Transformer的骨干网络（Swin-L和PVTv2-B5）能为模型带来更好的鲁棒性。然而，对于需要更复杂流程的任务，例如全景分割，我们认为任务头的设计也对模型的鲁棒性起着重要作用。例如，Panoptic SegFormer (Swin-L)在COCO-C上的平均结果为47.2% PQ，比MaskFormer (Swin-L)高出5.5% PQ，这比它们在干净数据上的差距（2.9% PQ）更大。我们认为这是由于我们基于Transformer的掩码解码器比MaskFormer基于卷积的像素解码器具有更强的鲁棒性。

5. 结论

局限性。 本工作依赖可变形注意力来处理多尺度特征，速度稍慢。我们的模型仍然难以处理空间形状更大的特征，并且对于小目标表现不佳。

讨论。 最近，分割领域尝试使用统一的流程来处理各种任务，包括语义分割、实例分割和全景分割。然而，我们认为完全的统一在概念上令人兴奋，但不一定是合适的策略。考虑到各种分割任务之间的异同，“求同存异”是更合理的指导理念。通过查询解耦策略，我们可以在同一范式中处理可数物体和背景，因为它们都被表示为查询。此外，我们也可以为可数物体或背景设计定制化的流程。这种灵活的策略更适合各种分割任务。目前，针对特定任务的设计仍然能带来更好的性能。我们鼓励社区进一步探索统一的分割框架，并期望全景分割变换器能够启发未来的工作。

6. 致谢

本研究由国家自然科学基金项目61672273和项目61832008资助。罗平获得香港普通研究基金编号27208720和17212120的资助。王文海和卢通是通讯作者。

参考文献

[1] 尼古拉斯·卡里昂、弗朗西斯科·马萨、加布里埃尔·辛纳夫、尼古拉斯·乌苏尼尔、亚历山大·基里洛夫和谢尔盖·扎戈鲁伊科。使用Transformer模型的端到端目标检测。发表于欧洲计算机视觉会议，2020年。1, 2, 3, 4, 5, 6, 7[2] 王慧宇、朱宇坤、哈特维希·亚当、艾伦·尤尔和陈良杰。Max-Deeplab：使用掩码Transformer的端到端全景分割。发表于计算机视觉与模式识别会议，2021年。1, 2, 3, 4, 5, 6[3] 程博文、亚历山大·G·施温和亚历山大·基里洛夫。逐像素分类并非语义分割的全部。发表于神经信息处理系统大会，2021年。1, 2, 3, 5, 6, 7, 8[4] 张文伟、庞江淼、陈恺和罗成昌。K-Net：迈向统一的图像分割。发表于神经信息处理系统大会，2021年。1, 2, 3, 5, 6[5] 王文海、谢恩泽、李翔、范登平、宋凯涛、梁定、卢通、罗平和邵岭。PVTv2：使用金字塔视觉Transformer改进基线。arXiv:2106.13797，2021年。1, 6[6] 亚历山大·基里洛夫、何恺明、罗斯·吉尔希克、卡斯滕·罗瑟和彼得罗·佩罗纳。全景分割。发表于计算机视觉与模式识别会议，2019年。1, 2, 4, 5[7] 亚历山大·基里洛夫、罗斯·吉尔希克、何恺明和彼得罗·佩罗纳。全景特征金字塔网络。发表于计算机视觉与模式识别会议，2019年。1, 2, 3, 6[8] 纳瓦尼特·博德拉、巴拉特·辛格、拉马·切拉帕和拉里·S·戴维斯。Soft-NMS——用一行代码改进目标检测。发表于IEEE国际计算机视觉会议论文集，第5561–5569页，2017年。1[9] 熊宇文、廖仁杰、赵恒爽、胡锐、白敏、埃尔辛·尤默和拉克尔·乌尔塔松。UPSNet：统一的泛视分割网络。发表于计算机视觉与模式识别会议，2019年。1, 2[10] 林宗仪、彼得罗·佩罗纳、罗斯·吉尔希克、何恺明、巴拉特·哈里哈兰和谢尔盖·贝隆吉。用于目标检测的特征金字塔网络。发表于计算机视觉与模式识别会议，2017年。1[11] 林宗仪、迈克尔·梅尔、谢尔盖·贝隆吉、詹姆斯·海斯、彼得罗·佩罗纳、德瓦·拉马南、彼得罗·佩罗纳和C·劳伦斯·齐特尼克。微软COCO：上下文中的常见物体。发表于欧洲计算机视觉会议，2014年。2, 5[12] 朱锡洲、苏伟杰、陆乐威、李斌、王晓刚和戴继峰。可变形DETR：用于端到端目标检测的可变形Transformer。发表于国际学习表征会议，2020年。2, 3, 4, 6, 7[13] 方宇新、杨树生、王兴刚、李玉、方晨、单莹、冯斌和刘文予。实例作为查询。发表于IEEE/CVF国际计算机视觉大会（ICCV），第6910–6919页，2021年10月。2, 3, 6[14] 陈恺、庞江淼、王家齐、熊宇、李晓晓、孙书阳、冯万森、刘子纬、史建平、欧阳万里等。用于实例分割的混合任务级联。发表于计算机视觉与模式识别会议，2019年。2, 6[15] 乌杰瓦尔·邦德、巴勃罗·F·阿尔坎塔里利亚和斯特凡·洛伊滕埃格。迈向无边界框的全景分割。发表于德国模式识别会议，2020年。2

[16] 李奇竹、齐晓娟和菲利普·H·S·托尔。统一全景分割的训练与推理。载于IEEE/CVF计算机视觉与模式识别会议论文集，第13320–13328页，2020年。[17] 程博文、麦克斯韦·D·柯林斯、朱宇坤、刘婷、黄煦涛、哈特维希·亚当和陈良杰。全景DeepLab：一个简单、强大且快速的底部全景分割基线。载于CVPR，2020年。2[18] 杨天如、麦克斯韦·D·柯林斯、朱宇坤、黄志靖、刘婷、张骁、施薇薇、乔治·帕潘德鲁和陈良杰。Deeperlab：单次图像解析器。arXiv:1902.05093，2019年。2[19] 高乃宇、单彦虎、王玉培、赵鑫、余一楠、杨明和黄凯奇。SSAP：具有亲和力金字塔的单次实例分割。载于ICCV，2019年。2[20] 李彦伟、陈新泽、朱征、谢凌曦、黄冠、杜大龙和王新刚。注意力引导的统一网络用于全景分割。载于CVPR，2019年。2[21] 李彦伟、赵恒爽、齐晓娟、王立威、李泽铭、孙剑和贾佳亚。用于全景分割的全卷积网络。载于CVPR，2021年。2, 3, 6, 7, 8[22] 田植、沈春华和陈浩。用于实例分割的条件卷积。载于ECCV，2020年。2, 4[23] 何恺明、张祥雨、任少卿和孙剑。用于图像识别的深度残差学习。载于CVPR，2016年。3, 6[24] 阿希什·瓦斯瓦尼、诺姆·沙泽尔、尼基·帕尔马、雅各布·乌斯佐凯特、利昂·琼斯、艾丹·N·戈麦斯、乌卡什·凯泽和伊利亚·波洛苏欣。注意力就是一切。载于NeurIPS，2017年。3[25] 董斌、曾凡高、王天财、张祥雨和韦毅辰。SOLQ：通过学习查询分割对象。NeurIPS，2021年。3, 6[26] 任少卿、何恺明、罗斯·吉尔希克和孙剑。Faster R-CNN：通过区域提议网络实现实时目标检测。NeurIPS，2015年。3[27] 林宗仪、普里亚·戈亚尔、罗斯·吉尔希克、何恺明和皮奥特·多拉尔。用于密集目标检测的Focal损失。载于ICCV，2017年。3, 5[28] 王新龙、张汝峰、孔涛、李磊和沈春华。SOLOv2：动态且快速的实例分割。NeurIPS，2020年。3, 4, 5, 6[29] 王新龙、孔涛、沈春华、蒋宇宁和李磊。SOLO：按位置分割对象。载于ECCV，2020年。4[30] 拉塞尔·斯图尔特、米哈伊洛·安德里卢卡和吴恩达。拥挤场景中的端到端行人检测。载于CVPR，2016年。4[31] 哈罗德·W·库恩。用于分配问题的匈牙利算法。海军研究物流季刊，2 97，1955年1 2 83 5（-）：–。

[32] 福斯托·米莱塔里、纳西尔·纳瓦布和赛义德-艾哈迈德·艾哈迈迪。V-网络：用于体积医学图像分割的全卷积神经网络。载于国际三维视觉会议（3DV），2016年。5[33] 周博磊、赵航、泽维尔·普伊格、桑贾·菲德勒、阿德拉·巴里乌索和安东尼奥·托拉尔巴。通过ADE20K数据集进行场景解析。载于IEEE计算机视觉与模式识别会议论文集，第633–641页，2017年。5, 6[34] 刘泽、林雨桐、曹越、胡涵、魏一轩、张正、林史蒂芬和郭柏宁。Swin Transformer：使用移位窗口的分层视觉Transformer。ICCV，2021年。6[35] 陈崇松、任佳伟、金岱升、蔡忠昂、余存俊、王柏润、张明远和吴金毅。ICCV 2019联合COCO与Mapillary研讨会：COCO全景分割挑战赛赛道技术报告：具有类别引导融合的全景HTC。SHR，56(84.1):67–2。6[36] 李彦伟、赵恒爽、齐晓娟、陈宇康、齐璐、王立威、李泽铭、孙剑和贾佳亚。具有基于点监督的全景分割全卷积网络。arXiv预印本 arXiv:2108.07682，2021年。6[37] 吴阳欣、张耕玮、高益明、邓夏军、龚柯、梁小丹和林倞。用于全景分割的双向图推理网络。载于IEEE/CVF计算机视觉与模式识别会议论文集，第9080–9089页，2020年。6[38] 吴阳欣、张耕玮、徐航、梁小丹和林倞。Auto-Panoptic：用于全景分割的协作式多组件架构搜索。神经信息处理系统进展，33，2020年。6[39] 马宁宁、张祥雨、郑海涛和孙剑。ShuffleNet V2：高效CNN架构设计的实用指南。载于欧洲计算机视觉会议论文集（ECCV），第116–131页，2018年。6[40] 何恺明、乔治亚·吉奥克萨里、皮奥特·多拉尔和罗斯·吉尔希克。掩码区域卷积神经网络。载于ICCV，2017年。6[41] 程博文、罗斯·吉尔希克、皮奥特·多拉尔、亚历山大·C·伯格和亚历山大·基里洛夫。边界交并比：改进以对象为中心的图像分割评估。载于IEEE/CVF计算机视觉与模式识别会议论文集，第15334–15342页，2021年。6, 7[42] 克里斯托夫·卡曼和卡斯滕·罗瑟。语义分割模型的鲁棒性基准测试。载于IEEE/CVF计算机视觉与模式识别会议论文集，第8828–8838页，2020年。8[43] 谢恩泽、王文海、于志定、阿尼玛·阿南德库马尔、何塞·M·阿尔瓦雷斯和罗平。Segformer：使用Transformer进行语义分割的简单高效设计。载于NeurIPS，2021年。8[44] 斯里纳德·博贾纳帕利、阿扬·查克拉巴蒂、丹尼尔·格拉斯纳、李大亮、托马斯·温特希纳和安德烈亚斯·法伊特。理解用于图像分类的Transformer模型的鲁棒性。arXiv预印本 arXiv:2103.14586，2021年。8[45] 穆扎马尔·纳西尔、坎查纳·拉纳辛格、萨尔曼·汗、穆纳瓦尔·哈亚特、法哈德·沙赫巴兹·汗和杨明玄。视觉Transformer的有趣特性。 *arXiv*预印本 *arXiv:2105.10497*，2021年。 8