用于语义分割的低分辨率自注意力

Yu-Huan Wu, Shi-Chen Zhang, Yun Liu, Le Zhang, Xin Zhan, Daquan Zhou,
Jiashi Feng, Ming-Ming Cheng, and Liangli Zhen

摘要— 语义分割任务天然地需要像素级分割所需的高分辨率信息和类别预测所需的全局上下文信息。虽然现有的视觉 Transformer 模型展现出良好的性能，但它们通常依赖高分辨率上下文建模来实现这一性能，这会显著增加计算量。针对这一问题，本文打破了传统方法的限制，提出了低分辨率自注意力（Low-Resolution Self-Attention, LRSA）机制，该机制通过显著降低计算成本（即 FLOPs）来捕获全局上下文信息。具体而言，本文的方法在固定低分辨率空间内计算自注意力（不受输入图像分辨率影响），同时通过额外的 3×3 深度卷积在高分辨率空间中捕获细节信息。为了验证 LRSA 机制的有效性，本文设计了具有编码器-解码器结构的视觉 Transformer 模型 LRFormer。本文在 ADE20K、COCO-Stuff 和 Cityscapes 等数据集上进行了广泛的实验，结果表明 LRFormer 的性能超过了当前最先进的模型。相关代码已开源，详见：https://github.com/yuhuan-wu/LRFormer。

关键词—低分辨率自注意力，语义分割，视觉 Transformer

1 简介

语义分割（semantic segmentation）[2]–[4] 作为计算机视觉领域的一个基本问题，其目的在于为图像中的每个像素分配语义类别标签。现有的语义分割模型 [5], [6] 通常依赖于预训练的骨干网络（backbone network）[7], [8] 进行特征提取，并在此基础上采用专门设计的模块实现像素级预测。过去十年来，骨干网络在特征提取能力上的突破也极大地推动了语义分割领域的发展 [9]–[11]。本文以一个全新的视角出发，对语义分割中的特征提取方法进行改进。

现有的研究普遍认为，语义分割作为一种密集预测任务，需要高分辨率特征以确保准确性。相比之下，图像分类仅需低分辨率特征图（如输入分辨率的 1/32）即可完成类别预测。基于卷积神经网络（CNN）的语义分割模型通常通过减小骨干网络的步长（stride）来提高特征分辨率（例如提高至 1/8）[12]–[15]。基于 Transformer 的语义分割方法同样延续了这一特性，进一步验证了高分辨率特征在语义分割中的必要性。

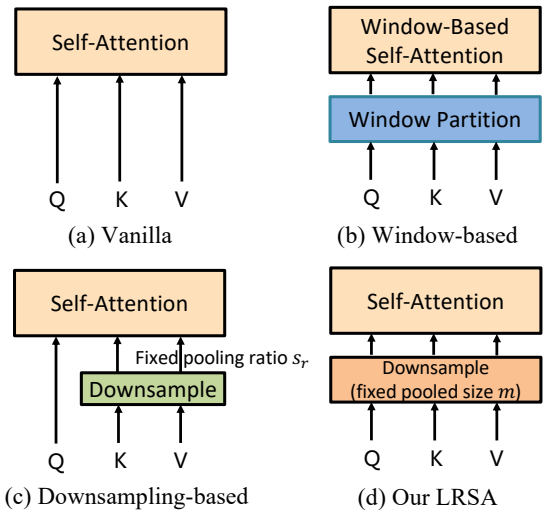


图 1. 现有视觉 Transformer 与本文所提出方法的自注意力计算范式对比。代表性方法包括：(a) ViT [16]、DeiT [17]；(b) Swin [18]、CSwin [19]；(c) PVT [20]、SegFormer [10]、P2T [21]；**以及** (d) **本文所提出的** LRFormer。**图中未绘制位置编码模块。**(d) **的更多细节可参见图**3。

高分辨率特征能够有效捕捉局部细节，而上下文信息则有助于理解场景的整体语义。上下文特征可以揭示场景中不同组成部分之间的关系 [22]，从而缓解仅依赖局部特征可能导致的歧义。因此，大量研究工作 [2], [23] 致力于扩大 CNN 的感受野。相比之下，视觉 Transformer 由于使用自注意力（self-attention）机制，天然具备全局感受野，能够直接建模全局关系。然而，传统注意力机制的计算复杂度与输入序列长度的平方成正比，导致较高的计算成本。有趣的是，一些重要

- 吴宇寰（*Yu-Huan Wu*）最初在南开大学开展本研究工作。吴宇寰与甄亮利（*Liangli Zhen*）目前隶属于新加坡科技研究局高性能计算研究所（*Institute of High Performance Computing, IHPC*）。*(E-mail: wyh.nku@gmail.com, llzhen@outlook.com)*
- 张世辰（*Shi-Chen Zhang*）、刘云（*Yun Liu*）与程明明（*Ming-Ming Cheng*）隶属于 *NKIARI*（中国深圳福田）及南开大学天津视觉计算与智能感知实验室（*VCIP*）。*(E-mail: zhangshichen@mail.nankai.edu.cn, liuyun@nankai.edu.cn, cmm@nankai.edu.cn)*
- 张乐（*Le Zhang*）隶属于电子科技大学。*(E-mail: zhangleuestc@gmail.com)*
- 占新（*Xin Zhan*）隶属于有鹿机器人科技，中国杭州。*(E-mail: zhanxin@udeer.ai)*
- 周大权（*Daquan Zhou*）隶属于北京大学电子与计算机工程学院。*(E-mail: zhoudaquan21@gmail.com)*
- 冯佳时（*Jiashi Feng*）隶属于字节跳动（*Bytedance Inc.*），新加坡。*(E-mail: jshfeng@bytedance.com)*
- 程明明为通信作者。
- 本文是 *[1]* 的中译版，由朱子轩、吴宇寰进行翻译和校正。

研究 [10], [21], [24] 通过在自注意力计算中对部分特征（即，键（Key）和值（Value））特征进行适当下采样，显著降低了计算复杂度。

然而，本文注意到自注意力计算的开销对于现有视觉 Transformer 而言依然是一个不可忽视的瓶颈，这一点可从表12中得到印证。因此，本文希望进一步深入探讨 Transformer 核心组件——自注意力模块中的下采样问题。有别于以往仅对键和值特征进行下采样的方法 [10], [21], [24]，本文提出对查询（Query）、键和值全部特征同时进行下采样。以此方式，自注意力的输出将呈现为低分辨率形式，从而使 Transformer 主体特征维持在低分辨率状态。此外，本文采用固定的下采样尺寸而非下采样比例，以实现自注意力模块极低的计算复杂度。本文称该方法为**低分辨率自注意力（Low-Resolution Self-Attention，LRSA）**。

图1 对比了现有自注意力方法与本文提出的 LRSA（Low-Resolution Self-Attention）的差异。原始的自注意力机制（Vanilla self-attention）[16]（图1(a)）需在原始分辨率下计算全局特征关系，计算成本极高。基于窗口的方法 [18], [19], [25], [26]（图1(b)）通过将特征划分为局部窗口，仅在窗口内计算自注意力，从而减少计算量。基于下采样的方法 [10], [20], [21], [27]（图1(c)）保持查询（Query）的原始尺寸，而对键（Key）和值（Value）特征通过固定比例的池化进行下采样，其计算复杂度随输入分辨率线性增长。本文提出的 LRSA 方法（图1(d)）通过将查询、键和值统一下采样至固定低分辨率，实现了与输入分辨率无关的极低计算复杂度。具体复杂度分析见 §3.1。

尽管 LRSA 在高效捕获全局上下文信息方面具有显著优势，保持细粒度局部细节对于提升语义分割性能同样关键。为同时满足这两方面需求，本文在纯低分辨率空间中使用 LRSA 获取全局上下文信息，同时在高分辨率空间采用小卷积核（3×3）深度卷积捕获局部细节。基于这一设计原则，本文构建了新型骨干网络，并设计简单解码器用于聚合多层次特征以完成语义分割任务。该模型被命名为**低分辨率 Transformer（Low-Resolution Transformer，LRFormer）**。在 ADE20K [28]、COCO-Stuff [29] 和 Cityscapes [30] 等数据集上的实验结果表明（如，图2），LRFormer 系列模型性能优于当前最先进方法。

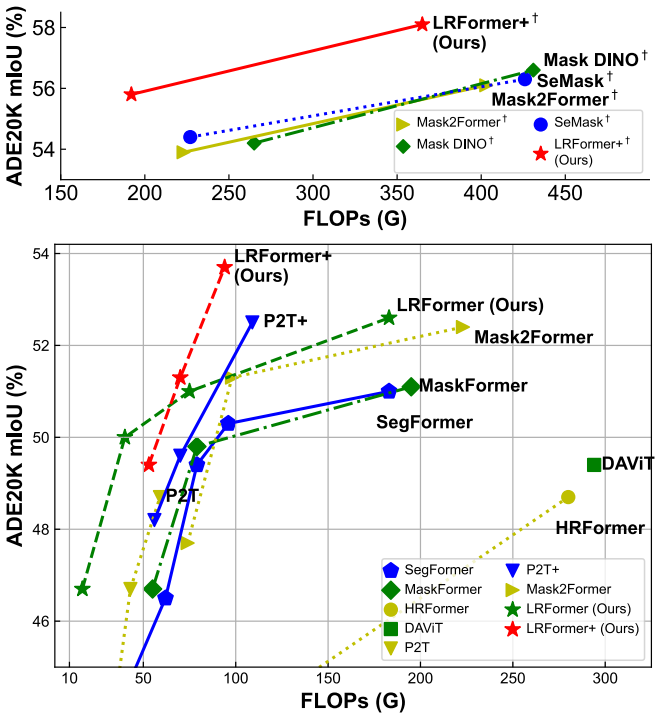


图 2. 在 ADE20K 数据集 [28] 上的实验对比。标注为 "†" 的方法表示其在 ImageNet-22K 数据集上进行了预训练。数据来源于表3和表13。

2 相关工作

2.1 语义分割

语义分割作为计算机视觉的基础任务，面临着目标尺寸、纹理和光照条件等多重现实挑战。FCN [31] 作为该领域的里程碑式工作，首次实现了基于卷积神经网络（CNN）的端到端语义分割。在此基础之上，后续研究主要集中在多尺度特征表示优化 [5], [6], [32]、边缘感知能力增强 [33]–[36]、上下文信息建模 [22], [37] 和视觉注意力机制引入 [3], [4], [9], [12], [15]。这些研究通过改进 FCN [31] 的语义头（semantic head）设计，取得了显著进展。其中许多方法 [2]–[6], [9], [12]–[15], [38]–[40] 普遍利用高分辨率特征，通常在输入分辨率的 1/8 尺度进行预测以确保精度。还有部分方法基于查询的解码框架 [41]–[46]，其创新性地重构了解码器设计范式。如，Panoptic SegFormer [41] 通过查询解耦策略分别处理"thing"和"stuff"类别；MaskFormer [42] 开创了基于 Transformer 的掩码分类方法；Mask2Former [43] 引入多尺度掩码注意力机制；K-Net [44] 利用动态核函数统一多任务分割；K-means Transformer [45] 则通过融合 K-means 聚类优化掩码生成。

近年来，视觉 Transformer [16] 在语义分割任务中的卓越表现已得到广泛验证 [10], [11], [42], [43], [47]。其性能提升主要源于视觉 Transformer 特有的全局建模能力，这一特性与语义分割任务对全局上下文理解的核心需求高度契合。例如，SETR [47] 开创性地采用 ViT 作为编码器架构，结合多层特征聚合策略；SegFormer [10] 设计金字塔型 Transformer 编码器，配合轻量级 MLP 解码器；FeedFormer [48] 创新性地构建特征查询机制，显著提升结构信息建模能力。更多关于视觉 Transformer 在语义分割中的应用讨论详见 §2.3。

近年来出现了一些通用型或基础型的视觉大模型，致力于探索大规模建模能力，如，EVA [49]、OneFormer [50] 和 One-Peace [51] 等模型展现出强大的表征能力，在语义分割任务中取得了显著成效，其中部分模型甚至具备处理多模态输入的能力。需要说明的是，由于 GPU 计算资源限制，本研究暂未将 LRFormer 与上述大型视觉模型进行性能对比。

2.2 卷积神经网络（CNN）

鉴于基于 CNN 的语义分割模型依赖于 CNN 骨干网络进行特征提取，本文在此回顾一些具有代表性的 CNN 架构。自 AlexNet [52] 开创性的工作以来，学者们开发了多种技术以增强 CNN 的特征表达能力，并取得了显著成果。VGG [53]、GoogleNet [54]、ResNet [7] 和 DenseNet [55] 等代表性工作通过增加网络深度显著提升了特征表达能力。在此基础上，ResNeXt [8]、Res2Net [56] 和 ResNeSt [57] 等研究进一步探索了基数（cardinality）优化的设计空间，而 SENet [58] 和 SKNet [59] 则通过注意力机制实现了动态特征选择。近年来，若干研究表明使用大卷积核的 CNN 架构也具有强大的性能表现 [60]–[62]。为确保语义分割任务中的特征图具备较高分辨率，语义分割模型通常会减小 CNN 骨干网络的步长，并引入空洞卷积（dilated convolutions）[23] 以维持较大的感受野。在此动机下，HRNet [63] 被提出，用于直接学习高分辨率的 CNN 特征表示。尽管 CNN 在众多视觉任务中取得了成功，但其在建模全局与长距离依赖关系方面仍存在局限性，而这类关系对于语义分割任务而言至关重要。

2.3 视觉 Transformer

Transformer 最初在自然语言处理（NLP）领域中被提出 [64]。Transformer 通过使用多头自注意力机制（multi-head self-attention, MHSA），能够有效建模全局依赖关系，这一特性使其在需要全局场景理解的计算机视觉任务中展现出独特优势。为适应视觉数据特点，ViT [16] 首创性地将图像分割为 16×16 的词符（token），并利用 Transformer 结构对这些词符进行处理，在图像识别任务中取得了优于 CNN 的性能表现。近年来，金字塔型的视觉 Transformer [11], [18], [20], [21], [24], [27], [65], [66] 被广泛应用于图像识别任务，尤其在语义分割中表现出色。例如，PVT [20] 和 MViT [27] 通过对键和值特征进行下采样，构建了金字塔视觉 Transformer 结构。特别地，MViT [27] 在每个阶段的第一个模块中将查询的分辨率降低一半，而无需在阶段之间重复进行图像块嵌入操作。Liu 等人 [18] 提出了基于窗口的视觉 Transformer——Swin Transformer，通过引入滑动窗口机制建模局部与全局关系。Yuan 等人 [11] 设计了 HRFormer，利用视觉 Transformer 直接学习高分辨率特征以服务于密集预测任务。Xia 等人 [67] 提出了 DAT 模型，通过可变形注意力机制对键和值特征进行可变形采样。Wu 等人 [21] 通过层内金字塔池化策略，引入高效的多尺度自注意力机制。Liu 等人 [68] 则提出以分层的方式计算自注意力。更多相关方法可参阅综述文献 [69]。

尽管现有研究表明视觉 Transformer 在语义分割中表现出色，但现有研究仍普遍认为：为了使自注意力机制有效捕获上下文信息，高分辨率特征是不可或缺的。基于窗口的视觉 Transformer 方法 [18], [19], [25], [26] 通过在局部窗口内计算自注意力以降低计算复杂度，从而得以维持特征图的高分辨率。而基于下采样的视觉 Transformer 方法 [10], [20], [21], [24], [27], [68] 则在保持查询特征分辨率不变的同时，仅对键和值特征采用固定池化比例进行部分下采样。这种策略相比原始自注意力机制大幅降低了计算复杂度，使得模型可以保留高分辨率特征，但对于高分辨率输入而言，其仍然面临着较大的计算负担（见表12）。相较之下，本文认为在自注意力中保持高分辨率以获取上下文信息的必要性有待商榷，并通过提出基于 LRSA 的 LRFormer 模型，对这一问题进行了系统性的研究。在多个公开基准数据集上的优异性能表明，本文提出的 LRFormer 在语义分割任务中具有显著优势。

表 1
不同自注意力机制的对比。其中，N 表示展开后的特征长度，C 表示特征通道数。为简单起见，本文省略了常数因子，例如窗口方法中的窗口大小，以及 LRSA 中的下采样尺寸。

模式	全局	空间相关性	复杂度
基于窗口划分的 [18]	✘	✔	$O(NC^2)$
基于因子分解的 [70]	✔	✘	$O(NC^2)$
基于下采样的 [10]	✔	✔	$O(N^2C + NC^2)$
LRSA (本文)	✔	✔	$O(NC + C^2)$

3 方法

本节首先在 §3.1中介绍低分辨率自注意力机制（Low-Resolution Self-Attention，LRSA）。接着在 §3.2中，本文基于 LRSA 构建了用于语义分割的低分辨率 Transformer（Low-Resolution Transformer，LRFormer）。LRFormer 的解码器部分将在 §3.3中进行详细说明。最后，本节在 §3.4中给出了本文方法的实现细节。

3.1 低分辨率自注意力

现有视觉 Transformer 通常致力于在自注意力过程中保持高分辨率的特征图，而本文提出的 LRSA 在低分辨率空间中计算自注意力，显著降低了计算成本。在详细介绍 LRSA 机制之前，本文先简要回顾视觉 Transformer 的基本架构。

Transformer 中自注意力机制的回顾。 视觉 Transformer [16] 已被证明在计算机视觉任务中具备强大的性能 [11], [18]–[21], [24]–[27]。其核心包含两个部分：多头自注意力机制（Multi-Head Self-Attention，MHSA）和前馈神经网络（Feed-Forward Network，FFN）。本文首先对 MHSA 进行详细介绍。给定输入特征 F_{in}，通过线性变换可分别获查询 Q、键 K 和值 V。标准多头自注意力机制（vanilla MHSA）可表示为：

$$\text{Attention}(F_{in}) = \text{Softmax}(\frac{QK^T}{\sqrt{d_k}})V, \tag{1}$$

其中，d_k 表示 F_{in} 的通道数。为简明起见，这里省略了多头计算的细节。标准自注意力机制的整体计算复杂度为 $O(N^2C +$

NC^2)，其中 N 是词符数量，C 是通道数，$F_{in} \in \mathbb{R}^{N\times C}$。由于自然图像中的词符数量通常较大，因此标准自注意力机制的计算开销非常高。

现有的解决方案。 为在保持特征图高分辨率的同时降低计算成本，近年来的下采样式视觉 Transformer 方法 [10], [20], [21], [24], [27] 将自注意力的计算方式修改为：

$$\text{Attention}(F_{in}) = \text{Softmax}(\frac{QK_s^T}{\sqrt{d_k}})V_s, \tag{2}$$

其中，K_s 和 V_s 分别表示键 K 和值 V 经过固定的下采样比例 s_r 得到的下采样特征。为简便起见，此处省略了 $1D \leftrightarrow 2D$ 的特征变换过程。K_s 和 V_s 的序列长度约为原始 K 和 V 的 $1/s_r^2$。然而，如果原始的 K 和 V 长度本身就很大，即使经过下采样，K_s 和 V_s 仍然可能是较长的序列，进而在自注意力计算中引入大量的计算开销。本段仅介绍与本文方法最相关的下采样类 Transformer。

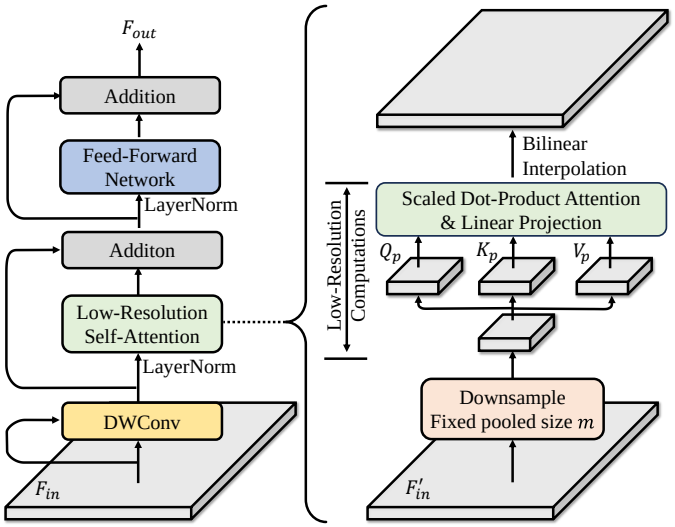
本文的解决方案。 不同于传统方法，本文从全新视角解决标准自注意力的高计算量问题：本文不再保持特征图的高分辨率，而是在极低分辨率空间中处理特征（见图3(b)）。具体而言，本文提出的 LRSA 首先将输入特征 F_{in} 下采样至固定大小 m，随后应用多头自注意力机制：

$$\text{Attention}(F_{in}) = \text{Softmax}(\frac{Q_pK_p^T}{\sqrt{d_k}})V_p, \tag{3}$$

其中 Q_p、K_p 和 V_p 由下采样后的 F_{in} 通过线性变换获得，并且其长度固定为 m，与输入 F_{in} 的分辨率无关。与标准自注意力及先前方案相比，本文的 LRSA 具有更低的计算开销；更短的词符长度也有助于注意力的优化。最后，为匹配原始 F_{in} 的尺寸，本文在自注意力计算后采用双线性插值进行上采样。

复杂度与特征分析。 与现有视觉 Transformer 中的自注意力机制相比，LRSA 在计算复杂度方面明显更低。本文在表1中总结了近年来主流自注意力机制与本文提出的 LRSA 的主要特征及其计算复杂度。表中的“空间相关性”指的是自注意力是否在空间维度上进行；部分因子分解类（factorized）Transformer 方法（如 CoaT [70]）为降低计算开销，会在通道维度上执行自注意力计算。从表1可以看出，其他方法往往在计算复杂度、全局感受野以及空间相关性三者之间需要有所取舍。而相比之下，本文提出的 LRSA 在这三方面均展现出显著优势。

接下来本节将分析 LRSA 的计算复杂度。为简便起见，以下分析中不考虑 1D↔2D 的特征重排过程。LRSA 首先将输入特征 $F_{in} \in \mathbb{R}^{N\times C}$ 通过二维池化操作下采样至固定大小 $m \times C$，该步骤的计算复杂度为 $O(NC)$。随后，LRSA 对下采样后的特征进行线性变换和自注意力计算，其中线性变换的计算复杂度为 $O(mC^2)$，自注意力计算的复杂度为 $O(m^2C)$。最后，上采样操作的计算量与下采样相同，仍为 $O(NC)$。因此，LRSA 整体的计算复杂度为 $O(NC + mC^2 + m^2C)$。由于 m 是一个与输入序列长度 N 无关的常数（例如 $m = 16^2$），本文可以将上述复杂度简化为 $O(NC + C^2)$，远低于现有的自注意力机制。



(a) Basic Block of LRFormer (b) Low-Resolution Self-Attention

图 3. 本文方法 LRFormer 的基本块示意图。本文在 LRSA 之前添加了一个带有残差连接的 3×3 深度卷积（Depth-Wise Convolution, DWConv），同时该卷积也被应用于前馈网络（FFN）中的两个线性层之间。

3.2 低分辨率 Transformer

本节通过引入所提出的 LRSA 机制，构建了用于语义分割任务的低分辨率 Transformer（下称 LRFormer）。其整体架构如图4所示，采用了典型的编码器-解码器结构。

编码器-解码器结构。 编码器以一张自然图像作为输入，首先将其下采样为原始尺寸的 1/4，该设置遵循了当前主流研究工作中的做法 [18], [20], [21], [24], [27]。编码器采用金字塔结构，由四个阶段组成，每个阶段包含多个堆叠的基本块（basic block）。在每个阶段之间，本文引入一次图像块嵌入（patch embedding）操作，将特征尺寸进一步缩小一半。最终，编码器提取出四个层级的特征图 F_1, F_2, F_3, F_4，其对应的下采样步长分别为 4、8、16 和 32。在解码阶段，本文首先将 F_2, F_3, F_4 统一调整至与 F_2 相同的分辨率，并进行拼接后压缩通道数。该融合特征随后被送入解码器，进行进一步的语义推理，并通过一个 1×1 卷积层输出最终的语义分割图。解码器的详细结构将在 §3.3中介绍。

基本块（Basic block）。 如图3所示，本文提出的 LRFormer 的基本块结构与现有的 Transformer 模块类似 [18], [20]，由一个自注意力模块和一个前馈网络（FFN）构成。FFN 通常由两个线性层组成，中间使用 GELU 激活函数 [71]。不同之处在于，本文将自注意力模块替换为本文提出的 LRSA 机制。由于 LRSA 在极低分辨率空间中进行计算，其计算复杂度不受输入分辨率的影响，始终保持一个较低的状态。然而，低分辨率空间可能会导致输入特征的空间局部性信息丢失。受近期研究工作的启发 [10], [21], [72]，本文在位置编码和 FFN 中进一步

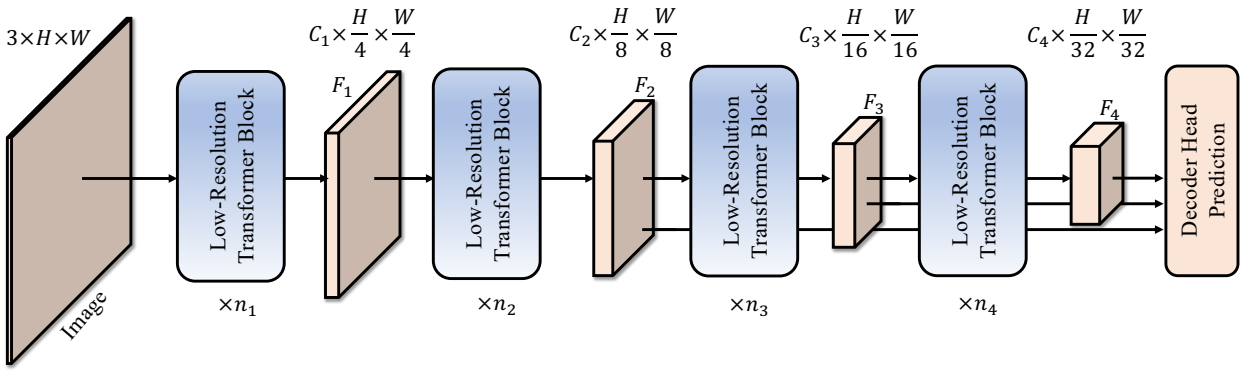


图 4. 本文提出的 LRFormer 流程示意图。特征图 F_2、F_3 和 F_4 将被送入解码器部用于语义分割任务。

引入了深度卷积（Depth-Wise Convolution, 下称 DWConv），以辅助捕获局部空间细节，从而增强特征提取能力。具体而言，本文在 LRSA 之前插入了一个带残差连接的 3×3 DWConv 层，以提供条件位置编码（conditional positional encoding）[72]。同样的策略也应用于 FFN 的两个线性层之间。因此，本文的基本块可被形式化地描述为：

$$\begin{aligned} F'_{in} &= F_{in} + \text{DWConv}(F_{in}), \\ F_{att} &= F'_{in} + \text{LRSA}(\text{LayerNorm}(F'_{in})), \\ F_{out} &= F_{att} + \text{FFN}(\text{LayerNorm}(F_{att})), \end{aligned} \tag{4}$$

其中，F_{in} 是输入特征，F_{att} 是 LRSA 的输出，F_{out} 是基本块的最终输出。由于 DWConv 的计算复杂度为 $O(NC)$，因此引入 DWConv 并不会改变 LRSA 整体保持在 $O(NC+C^2)$ 的低复杂度特性。

架构配置。 为了适配不同计算资源的需求，本文设计了四种不同规模的 LRFormer 变体，分别为 LRFormer-T/S/B/L/XL，通过在编码器的各个阶段堆叠不同数量的基本块实现结构上的差异。本文在表2中总结了各变体编码器的详细设置。在 ImageNet 预训练 [73] 阶段，LRFormer-T/S/B/L/XL 的计算开销分别与 ResNet-18 [7] 和 Swin-T/S/B/L [18] 相当。

3.3 解码器

在语义分割任务中，仅依赖编码器最终输出进行预测往往难以取得最优效果，因为多层级信息对于感知具有不同尺度和纵横比的目标非常重要 [10], [74]。因此，本文为 LRFormer 设计了一个结构简单但高效的解码器，用于有效整合多层次特征。值得注意的是，SegFormer [10] 等最新研究表明，使用 MLP 进行特征聚合也能获得较好的性能。然而，该方法未考虑来自不同层级特征之间的空间相关性。因此，本文在解码器中引入 LRSA 机制进行特征细化，从而增强 LRFormer 在语义层面的推理能力。

如前所述，本文将特征图 F_2、F_3、F_4 调整至与 F_2 相同的分辨率后进行拼接。随后，在拼接后的特征上施加一个 1×1 卷积，以压缩通道维度。接着，本文引入一个基本块（LRSA + FFN）对压缩后的特征进行细化。众所周知，来自编码器最高层的特征（即，F_4）通常包含最丰富的语义信息。为了避免在聚合高层特征（F_4）与低层特征（F_2、F_3）过程中丢失语义信息，本文将细化后的特征与 F_4 再次拼接，以进一步增强语义表达能力。之后，再接入一个基本块进行进一步的特征细化。最后，本文使用一个简单的 1×1 卷积对最终的细化特征进行预测，生成语义分割图。实验结果表明，如表10所示，本文所提出的结合 LRSA 的轻量解码器在语义分割任务中优于现有的主流解码器设计。

3.4 实现细节

在 LRFormer 中，本文采用了重叠式的图像块嵌入策略，即使用一个步长为 2 的 3×3 卷积操作，在各编码阶段之间将特征图下采样一半。为了以极小的计算开销增强 LRSA 的多尺度建模能力，本文在计算 LRSA 中的键和值特征时引入了金字塔池化策略（pyramid pooling）[21]，用于提取多尺度信息。在语义分割任务中，生成查询、键和值所需的固定下采样尺寸 m 设置为 16^2。而在 ImageNet 预训练阶段，考虑到 $m = 16^2$ 对图像分类任务而言过大，本文将其调整为 7^2。在解码器中，通道数分别设置为：LRFormer-T 为 256，LRFormer-S 为 384，LRFormer-B 为 512，LRFormer-L 为 640。

4 实验

4.1 实验设置

数据集。 本文在三个广泛使用的主流数据集上进行了实验。ADE20K [28] 是一个具有高度挑战性的场景解析数据集，包含 150 个语义类别，前景与背景内容丰富多样，分别包含 20K 张训练图像、2K 张验证图像和 3.3K 张测试图像。COCO-Stuff [29] 同时标注了 thing 类与 stuff 类，共包含 171 个细粒度的语义标签，数据集划分为 164K 张训练图像、5K 张验证图像、20K 张 test-dev 图像和 20K 张 test challenge 图像。Cityscapes [30] 是一个高质量的街景解析数据集，分别包含 3K 张训练图像、0.5K 张验证图像和 1.5K 张测试图像，适用

表 2
不同 LRFormer 变体（即 T/S/B/L/XL）的编码器详细配置。其中，C 表示特征通道数，C_h 表示每个注意力头的通道数，E 表示 FFN 的扩展倍率，n_i 表示第 i 个阶段的基本块数量。

阶段	输出大小	LRFormer-T	LRFormer-S	LRFormer-B	LRFormer-L	LRFormer-XL
1	$F_1: \frac{H}{4} \times \frac{W}{4}$	$C=48, E=8$ $C_h=24, n_1=2$	$C=64, E=8$ $C_h=32, n_1=3$	$C=80, E=8$ $C_h=40, n_1=4$	$C=96, E=8$ $C_h=48, n_1=4$	$C=128, E=8$ $C_h=64, n_1=4$
2	$F_2: \frac{H}{8} \times \frac{W}{8}$	$C=96, E=8$ $C_h=24, n_2=2$	$C=128, E=8$ $C_h=32, n_2=3$	$C=160, E=8$ $C_h=40, n_2=4$	$C=192, E=8$ $C_h=48, n_2=6$	$C=256, E=8$ $C_h=64, n_1=8$
3	$F_3: \frac{H}{16} \times \frac{W}{16}$	$C=240, E=4$ $C_h=24, n_3=6$	$C=320, E=4$ $C_h=32, n_3=12$	$C=400, E=4$ $C_h=40, n_3=15$	$C=480, E=4$ $C_h=48, n_3=18$	$C=640, E=4$ $C_h=64, n_3=22$
4	$F_4: \frac{H}{32} \times \frac{W}{32}$	$C=384, E=4$ $C_h=24, n_4=3$	$C=512, E=4$ $C_h=32, n_4=3$	$C=512, E=4$ $C_h=32, n_4=8$	$C=640, E=4$ $C_h=40, n_4=8$	$C=768, E=4$ $C_h=48, n_4=8$

于城市道路场景的理解任务。这些数据集涵盖了广泛的语义类别，为语义分割模型带来不同维度的挑战。

ImageNet 预训练。 本文采用主流的 *timm* 库来实现本文提出的网络。与其他主流方法一致，本文首先在 ImageNet-1K 数据集上对 LRFormer 的骨干网络进行预训练。ImageNet-1K 包含 130 万张训练图像和 5 万张验证图像，共涵盖 1000 个物体类别。在 ImageNet 预训练阶段，本文省略了解码器部分，仅对骨干网络进行训练。为规范训练过程，本文遵循了已有工作中常用的数据增强策略和优化方法 [10], [18], [75]。本文使用 AdamW 优化器 [76]，其初始学习率设为 0.001，权重衰减为 0.05，采用余弦学习率衰减策略，批大小为 1024。在预训练过程中不使用模型 EMA（Exponential Moving Average）。骨干网络训练 300 个 epoch，同时为缓解大模型过拟合问题，本文采用了 Layer Scale 机制 [77]，这一策略也已被近期研究广泛采用 [60], [77]。对于 LRFormer-L 模型，本文进一步遵循 [18], [60] 的做法，先在完整的 ImageNet-22K 数据集上训练 90 个 epoch，然后在 ImageNet-1K 数据集上微调 30 个 epoch。在微调阶段，学习率设为 5e-5，每个 mini-batch 包含 512 张图像。

语义分割训练。 本文使用 *mmsegmentation* 框架对网络进行语义分割训练。采用 AdamW [76] 作为默认优化器，初始学习率为 0.00006，权重衰减 0.01，并使用因子为 1.0 的 *poly* 学习率衰减策略。按照 [10], [18] 的做法，LayerNorm [83] 层的权重衰减系数设为 0。在数据增强方面，本文采用与 [10], [18] 相同的策略：首先对图像进行 0.5 ∼ 2 倍的随机缩放，然后随机水平翻转，最后分别对 ADE20K、COCO-Stuff 与 Cityscapes 数据集随机裁剪 512×512、512×512 和 1024×1024 的图像块。特别地，对于 ADE20K 数据集上的最大模型 LRFormer-L，保持 640×640 的裁剪尺寸以与最新工作保持一致。批大小分别设为 16、16 和 8，用于 ADE20K、COCO-Stuff 和 Cityscapes 数据集；训练迭代次数分别为 160K、80K 和 160K。训练过程中仅使用交叉熵损失，不额外引入辅助损失 [5] 或 OHEM [84] 等其他损失项。

表 3
在 ADE20K 数据集 [28] 上的最新方法比较。本文方法的结果用**加粗**标注。"†" 表示在 ImageNet-22K 上预训练的结果。

方法	FLOPs ↓	#Params ↓	mIoU ↑
SegFormer-B1 [10]	16G	14M	42.2%
Vim-Ti [78]	-	13M	41.0%
HRFormer-S [11]	109G	14M	44.0%
LRFormer-T (本文)	17G	13M	**46.7%**
SegFormer-B2 [10]	62G	28M	46.5%
P2T-Small [21]	43G	28M	46.7%
MaskFormer [42]	55G	42M	46.7%
FeedFormer-B2 [48]	43G	29M	48.0%
Mask2Former [74]	74G	47M	47.7%
LRFormer-S (本文)	40G	32M	**50.0%**
HRFormer-B [11]	280G	56M	48.7%
Vim-S [78]	-	46M	44.9%
SegFormer-B3 [10]	96G	47M	49.4%
LRFormer-B (本文)	75G	69M	**51.0%**
DPT-Hybrid [79]	308G	124M	49.0%
SegFormer-B5 [10]	183G	85M	51.0%
DAViT-B [80]	294G	121M	49.4%
FasterViT-4 [81]	323G	457M	49.1%
InternImage-B [82]	296G	128M	50.8%
MaskFormer [42]	195G	102M	51.3%
LRFormer-L (本文)	183G	113M	**52.6%**
SETR-MLA† [47]	-	302M	48.6%
MaskFormer† [74]	195G	102M	53.1%
CSWin-B† [19]	463G	109M	51.8%
LRFormer-L† (本文)	183G	113M	**54.2%**

语义分割测试。 在测试阶段，本文保持输入图像的原始纵横比，并将 ADE20K 与 COCO-Stuff 数据集的图像调整为短边 512、且长边不超过 2048。按照 [10] 的建议，在 ADE20K 数据集上对 LRFormer-L 的输入图像调整为短边 640、且长

表 4
在完整的 COCO-Stuff 数据集 [29] 上与最新基于 Transformer 的方法的比较。本文方法的结果用**加粗**标注。

方法	FLOPs ↓	#Params ↓	mIoU ↑
HRFormer-S [11]	109G	14M	37.9%
SegFormer-B1 [10]	16G	14M	40.2%
LRFormer-T (本文)	17G	13M	**43.9%**
SegFormer-B2 [10]	62G	28M	44.6%
LRFormer-S (本文)	40G	32M	**46.4%**
HRFormer-B [11]	280G	56M	42.4%
SegFormer-B3 [10]	79G	47M	45.5%
SegFormer-B5 [10]	112G	85M	46.7%
LRFormer-B (本文)	75G	69M	**47.2%**
LRFormer-L (本文)	122G	113M	**47.9%**

表 5
在 Cityscapes 数据集 [30] 上与最近基于 Transformer 的方法的比较。本文方法的结果用**加粗**标注。FLOPs 基于 1024×2048 的输入尺寸计算。

方法	FLOPs ↓	#Params ↓	mIoU ↑
HRFormer-S [11]	872G	14M	80.0%
SegFormer-B1 [10]	244G	14M	78.5%
LRFormer-T (本文)	122G	13M	**80.7%**
SegFormer-B2 [10]	717G	28M	81.0%
LRFormer-S (本文)	295G	32M	**81.9%**
HRFormer-B [11]	2240G	56M	81.9%
SegFormer-B3 [10]	963G	47M	81.7%
SegFormer-B5 [10]	1460G	85M	82.4%
LRFormer-B (本文)	555G	67M	**83.0%**
LRFormer-L (本文)	908G	111M	**83.2%**

边不超过 2560。对于 Cityscapes 数据集，本文同样遵循 [10]，采用 1024×1024 的滑动窗口裁剪策略进行测试。

表 6
ImageNet-1K 数据集 [73] 上的分类结果。本文方法的结果用**加粗**标注。带有 "†" 标记的结果表示在 ImageNet-22K 数据集上进行预训练。

模型	FLOPs ↓	#Params ↓	大小	Top-1 准确率 ↑
PVTv2-B1 [24]	2.1G	13M	224^2	78.7%
HAT-Net-T [68]	2.0G	13M	224^2	79.8%
P2T-Tiny [21]	1.8G	12M	224^2	79.8%
LRFormer-T (本文)	1.8G	13M	224^2	**80.8%**
Swin-T [18]	4.5G	28M	224^2	81.5%
MViTv2-T [86]	4.7G	24M	224^2	82.3%
Vim-S [78]	-	26M	224^2	81.4%
HAT-Net-S [68]	4.3G	26M	224^2	82.6%
ConvNeXt-T [60]	4.5G	29M	224^2	82.1%
LRFormer-S (本文)	4.7G	30M	224^2	**83.5%**
Swin-S [18]	8.7G	50M	224^2	83.0%
ConvNeXt-S [60]	8.7G	50M	224^2	83.1%
DAT-S [67]	9.0G	50M	224^2	83.7%
P2T-Large [21]	9.8G	55M	224^2	83.9%
LRFormer-B (本文)	9.3G	62M	224^2	**84.5%**
DeiT-B [17]	17.5G	86M	224^2	81.8%
RegNetY-16G [87]	16.0G	84M	224^2	82.9%
RepLKNet-31B [61]	15.3G	79M	224^2	83.5%
SwinT-B [18]	15.4G	88M	224^2	83.5%
ConvNeXt-B [60]	15.4G	89M	224^2	83.8%
FocalNet-B [88]	15.4G	89M	224^2	83.9%
CSwin-B [19]	15.0G	78M	224^2	84.2%
DAT-B [67]	15.8G	88M	224^2	84.0%
Vim-B [78]	-	98M	224^2	83.2%
LRFormer-L (本文)	15.7G	101M	224^2	**85.0%**
Swin-B† [18]	15.4G	88M	224^2	85.2%
ConvNeXt-B† [60]	15.4G	89M	224^2	85.8%
LRFormer-L† (本文)	15.7G	101M	224^2	**86.4%**
ConvNeXt-B† [60]	45.1G	89M	384^2	86.8%
Swin-B† [18]	47.0G	88M	384^2	86.4%
LRFormer-L† (本文)	46.3G	101M	384^2	**87.2%**
Swin-L† [18]	34.5G	197M	224^2	86.3%
ConvNeXt-L† [60]	34.4G	198M	224^2	86.6%
LRFormer-XL† (本文)	31.6G	187M	224^2	**87.0%**

4.2 对比

ADE20K。 实验结果见表3。本文的 LRFormer 与不同计算复杂度下最新 Transformer-based 及 Mamba-based 的方法进行了对比，其他方法的结果均来自其官方仓库。可以看出，本文的 LRFormer 展现出显著优势。以 mIoU 为准，LRFormer-T/S/B/L 分别比 SegFormer-B1/B2/B4/B5 [10], [18] 高出 4.5% / 3.5% / 2.6% / 1.6%。LRFormer-T 在 FLOPs 几乎减半的情况下，仍比 Swin-T-based Mask2Former [74] 高 2.3%。在 ImageNet-22K 预训练条件下，LRFormer 分别比最强的 Swin-B-based MaskFormer [18], [42] 及 UperNet-based CSwin [19], [85] 高 1.1% 和 2.4%，同时 FLOPs 更低。与采用线性复杂度的代表性 Mamba-based 模型 Vim [78] 相比，本文的 LRFormer 仍显著优于其性能。图2 所示的精度-FLOPs 可视化进一步直观展示了这种对比。

COCO-Stuff. 本文在表4中详细给出了实验结果。本文在不同规模的网络下评估了本文方法，并与近期的主流方法进行了比较。LRFormer 在所有规模上均取得了最高的 mIoU，全面超越其他方法。具体而言，LRFormer-T 的 mIoU 为 43.9%，比 HRFormer-S 高 3.7%，亦比 SegFormer-B1 高 3.7%。同样，LRFormer-S 和 LRFormer-B 分别较对应的 SegFormer 模型提升了 1.8% 和 1.7%。LRFormer-L 则比 SegFormer-B5 提升 1.2%。这些实验对比进一步证明了 LRFormer 在 COCO-Stuff 数据集上的优越性。

表 7
LRSA 固定池化尺寸设置的实验。实验结果表明，当池化尺寸超过 16×16 时，性能趋于饱和。

池化大小↓	FLOPs↓	训练时显存占用↓	mIoU↑
4×4	38G (-5%)	3.9GB (-7%)	46.3%
8×8	38G (-5%)	4.0GB (-5%)	46.8%
16×16	40G	4.2GB	48.5%
32×32	52G (+30%)	5.3GB (+26%)	48.6%
48×48	74G (+85%)	7.4GB (+76%)	48.7%
64×64	108G (+170%)	10.9GB (+160%)	48.5%

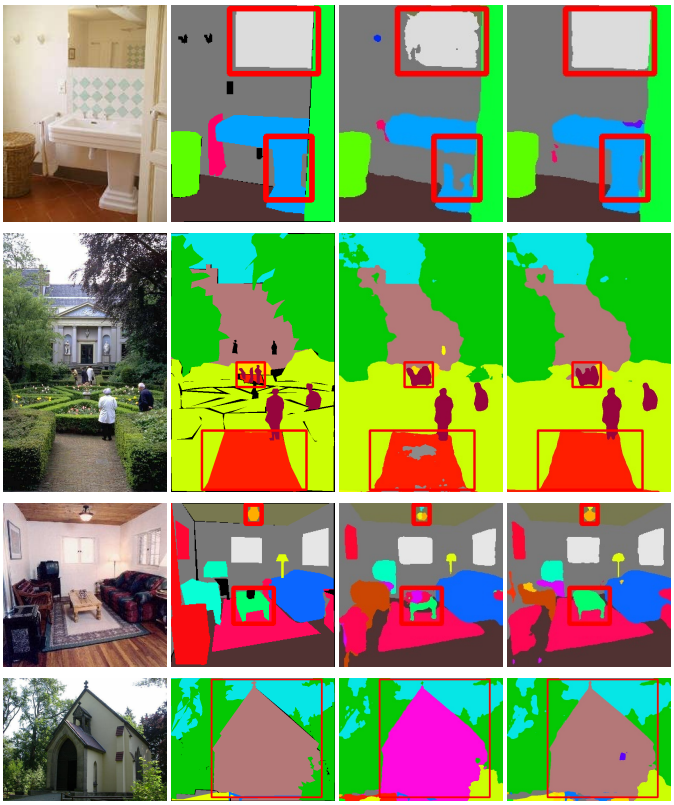


图 5. ADE20K 验证集上的定性可视化结果。从左至右依次为输入图像、真实标注、SegFormer [10] 的分割结果以及本文 LRFormer 的分割结果。红色框突出显示了显著改善的区域。

Cityscapes. 表5展示了本文 LRFormer 与近期主流方法在 Cityscapes 数据集上的实验对比。LRFormer 全面优于 SegFormer 和 HRFormer。可以观察到，由于输入分辨率较大，其余方法的 FLOPs 远高于本文模型。例如，SegFormer-B2 的计算量达到 717G FLOPs，而本文的 LRFormer-S 仅用 41% 的 FLOPs，mIoU 仍提升 0.9%。更多关于计算复杂度的分析见表12。

ImageNet。 鉴于本文已在 ImageNet 上预训练骨干编码器，本文也在 ImageNet 分类任务上对网络进行了评估，仅作参考。相关结果列于表6。本文将方法分为五组：前四组按计算量（约 2G、4.5G、9G、16G FLOPs）划分，第五与第六组为在

表 8
16×16 池化尺寸（默认）与较小池化尺寸（4×4）的性能比较。

类型	指标	默认	较小	相对变化
小	mIoU	36.2%	33.7%	-7.1%
	mAcc	46.3%	42.3%	-8.8%
中	mIoU	48.1%	45.3%	-5.7%
	mAcc	59.7%	56.8%	-4.9%
大	mIoU	57.2%	53.8%	-6.0%
	mAcc	68.3%	64.3%	-5.9%

ImageNet-22K 上预训练的结果。本文的 LRFormer 骨干编码器性能优于近期最先进的 CNN 方法，如 ConvNeXt [60] 和 RepLKNet [61]，以及基于 Transformer 的方法，如 DAT [67] 和 P2T [21]。

4.3 可视化分析。

为了直观展示本文方法的有效性，本文选取 SegFormer [10] 作为对比模型，并分别在 ADE20K 验证集与 Cityscapes 验证集上进行可视化比较，如图5和图6所示。结果表明，LRFormer 能够生成更加精确的分割图，尤其是在红色框标注的区域。本文发现，LRFormer 在保持目标分割完整性及捕获细粒度细节方面具有显著优势。

4.4 消融实验

本节通过一系列消融实验对本文的 LRFormer 进行深入分析。在本节的实验中，除非特别说明，均使用以下统一设置：选择 LRFormer-S 作为基线模型，在 8 张 GPU 上分别完成分类与语义分割训练。对于分类任务，在 ImageNet-1K 数据集上训练 100 个 epoch，对于语义分割任务，在 ADE20K 数据集上训练 80K 次迭代。其余超参数与 §4.1 中的配置保持一致。

固定池化尺寸。 本文在 ADE20K 语义分割任务中展示了实验结果（见表7）。对于每个基础块，当特征图尺寸小于设定的池化尺寸时，将省略池化操作。语义分割任务中默认的固定池化尺寸 m 为 16^2。实验结果表明，当池化尺寸较大（$m \geq 16^2$）时，性能已趋于饱和。与池化尺寸 8^2 相比，默认设置仅增加约 5% 的训练显存开销和 FLOPs。将池化尺寸进一步减小至 4^2 并不会显著提升效率；而当池化尺寸增大至 $32^2, 48^2, 64^2$ 时，在 ADE20K 语义分割上的性能仅有微弱提升甚至出现下降，同时 FLOPs 与训练显存开销显著增加（26% ∼ 170%）。在 Cityscapes 数据集上，本文进行了与上述内容相同的实验。该数据集的输入尺寸更大（1024×1024）。实验结果显示，LRFormer-L 在池化尺寸为 16^2 与 32^2 时的 mIoU 均为 83.2%，说明默认池化尺寸同样适用于更高分辨率的输入。尽管根据输入分辨率调整池化尺寸可能在小目标上保留更多细节，但本文的固定尺寸设计已能在各类目标尺度上表现良好。高分辨率 DWConv 分支与多层级特征聚合有效地在空间分辨率降低的情况下保留了小目标信息。此外，理

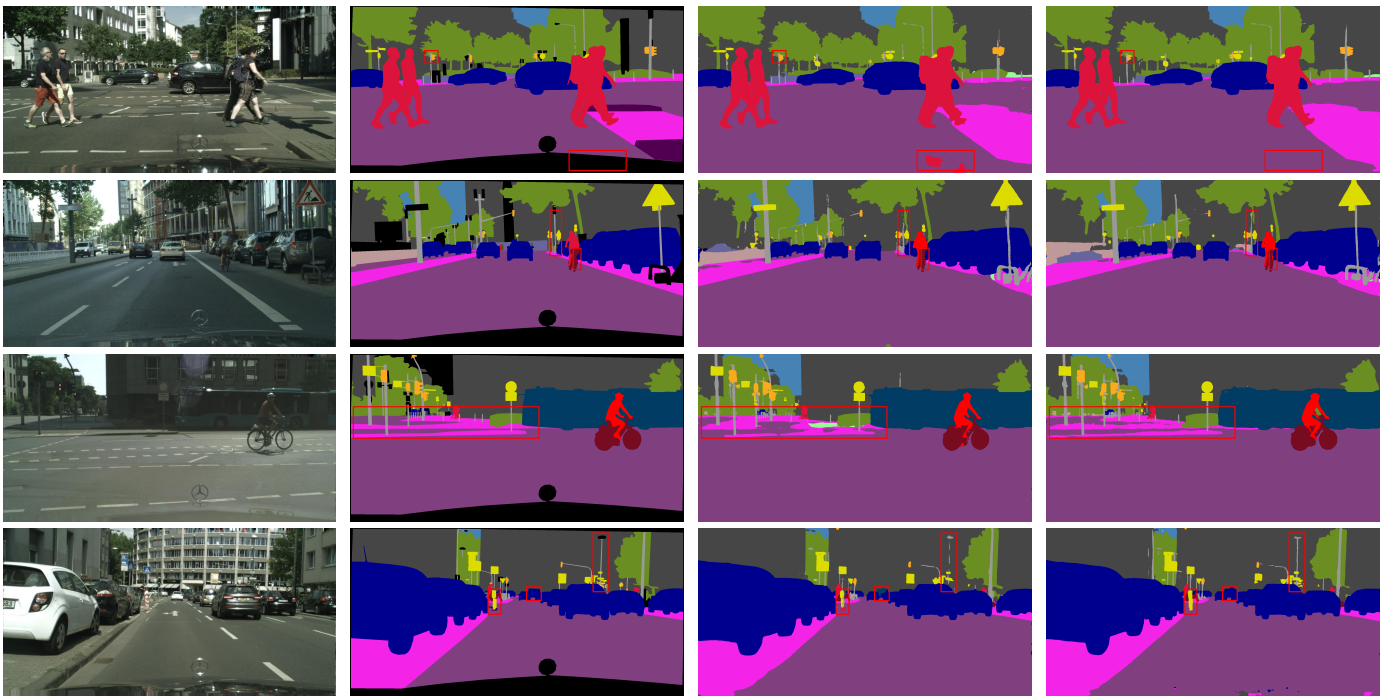


图 6. **Cityscapes 验证集上的定性可视化结果**。从左至右依次为输入图像、真实标注、SegFormer [10] 的分割结果以及本文 LRFormer 的分割结果。红色框突出显示了显著改善的区域。

表 9
关于空间局部性捕获的消融实验。

方法	显存	Top-1 Acc. ↑	mIoU ↑
LRFormer-S	14.5GB	81.6%	48.5%
w/o DWConv (bef. LRSA)	13.8GB	81.4%	48.0%
w/o DWConv (FFN)	11.7GB	81.1%	47.1%
w/o DWConv (Both)	11.0GB	80.4%	44.7%

表 10
本文简单解码器与其他主流解码器的比较。

Decoder Head	FLOPs ↓	#Params ↓	mIoU ↑
Ours	40G	32M	49.5%
w/ OCR [22]	48G	34M	48.0%
w/ PPM [5]	82G	44M	48.4%
w/ DA [3]	94G	42M	48.9%
w/ CC [9]	84G	42M	48.6%

想的骨干网络设计范式 [7], [18], [20], [60] 强调各阶段基本块应保持一致设置（在本文网络中即相同的池化尺寸），这种架构简洁性提高了实现效率，并减少了为各阶段分别寻找最优参数的工作量。综合考虑性能、FLOPs、训练显存以及理想的骨干设计范式，本文默认采用固定低分辨率池化尺寸。

局部信息捕获。 本文的 LRSA 仅在低分辨率空间中计算注意力。为了获得更精细的语义分割图，引入空间局部性信息的 3×3 深度可分离卷积（DWConv）是有帮助的。在 表9 中，本文分析了在 LRSA 之前和 FFN 中分别添加这两个 DWConv 的影响。由于这三种配置的 GFLOPs 几乎相同，这里不再列出具体数值。实验结果表明，当在 FFN 中的 LRSA 之前加入 DWConv 时，本文的 LRFormer 在 ADE20K 上的性能分别提升了 0.5% 和 1.4%，但训练显存开销相应增加了 5% 和 24%。若同时移除这两个 DWConv，与仅移除 FFN 中 DWConv 的情况相比，mIoU 将进一步下降 2.4%，训练显存则减少约 0.7 GB。这表明捕获局部信息在 LRFormer 中至关重要。基于此，本文在 LRFormer 中保留了这两个 DWConv 层。

小目标性能。 为了研究池化尺寸对不同尺寸目标的影响，本文将 ADE20K 的语义类别划分为小、中、大三类。离散目标依据其在现实世界中的典型尺寸进行分类，而非离散区域（如，天空）因在图像中通常占据较大空间，被归入大类别。结果如表8所示。将默认池化尺寸替换为更小的 4×4 会在所有类别上导致性能下降，其中小目标受到的影响最为显著。这一现象进一步证明，当特征被下采样到过低分辨率（如 4×4）时，特别是对于小目标，关键语义信息将会丢失。

不同解码器的比较。 本文提出的解码器通过借助 LRSA，在生成多层级特征的语义分割图时可以做到兼顾效率与性能。为验证 LRSA 的作用，本文将其与若干流行的解码器进行对比。这些解码器最初针对 CNN 设计，其输出特征图通常为原图的 1/8 分辨率；而本文的 LRFormer 骨干编码器输出的特征分辨率为原图的 1/32。为保证公平，本文首先对编码器最后几个阶段的特征进行上采样并拼接，然后将其输入这些流行解码器，其余流程保持一致。表10 给出了在 ADE20K 语义

表 11
关于解码器维度的讨论。当解码器的维度超过 384 时，性能将趋于饱和甚至下降。

维度	FLOPs ↓	#Params ↓	mIoU ↑
128	27G	30M	47.8%
256	32G	31M	49.2%
384	40G	32M	49.5%
512	50G	37M	49.6%
768	79G	46M	49.2%
1024	117G	58M	49.2%

表 12
不同输入尺寸下的内存占用与 FLOPs 分析。“Att. FLOPs”表示 MHSA 与上采样操作的 FLOPs 之和。“Memory”指语义分割训练过程中的显存占用。

方法	大小，批大小	显存 ↓	FLOPs ↓	Att. FLOPs ↓
LRFormer-S	512×512, 2	4.2GB	40G	0.8G
SegFormer-B2	512×512, 2	7.2GB	62G	3.4G
LRFormer-S	1024×1024, 1	5.7GB	145G	0.9G
SegFormer-B2	1024×1024, 1	18.8GB	279G	54.0G
LRFormer-S	1536×1536, 1	15.3GB	319G	1.1G
SegFormer-B2	1536×1536, 1	OOM	802G	293.6G

分割任务上的比较结果，骨干在 ImageNet-1K 上预训练 300 个 epoch。与 PPM [5]、DA [3] 和 CC [9] 相比，LRFormer 分别提升了 1.1%、0.6% 和 0.9%，且 FLOPs 不到对方的 50%。与 OCR [22] 相比，LRFormer 在仅耗费 83% FLOPs 的情况下取得了 1.5% 的性能增益。因此，默认设置下的 LRFormer 解码器在效率与效果上均优于其他流行解码器。

双线性插值。 尽管自注意力在低分辨率空间中计算，为满足残差连接的尺寸一致性需求，仍需对特征进行双线性插值放大。实验结果表明，在输入尺寸为 512^2 的情况下，采用 LRFormer-S 时，该双线性插值仅引入 0.1 ms 的延迟，占整体网络时延的 0.8%，几乎可以忽略。

不同骨干网络的解码器。 在本部分，本文在 SegFormer-B2 中用本文的解码器进行替换实验。结果表明，采用本文解码器的 SegFormer-B2 实现了 47.3% 的 mIoU，相比原始 SegFormer-B2 提高了 0.8%，同时计算量减少 28 GFLOPs。因此，将 SegFormer-B2 的解码器替换为本文解码器可在性能和效率方面带来显著提升。

解码器的维度。 为了在性能与计算成本之间取得最佳平衡，本文在将拼接后的多层级特征送入解码器之前，采用 1×1 卷积来压缩其通道维度。本文对多种通道维度设置进行了实验，并将结果汇总于 表11。实验中使用的骨干网络在 ImageNet-1K 上预训练了 300 个 epoch。实验结果表明，当通道维度设为 512 时性能最优；然而，将维度调整为 384 仅使 mIoU 降低 0.1%，却能节省 25% 的 FLOPs。因此，本文在 LRFormer-S 中将解码器的通道维度设定为 384，以实现性能与计算成本的最优折中。

表 13
在 ADE20K 数据集 [28] 上与最新查询式框架的比较。本文方法的结果以**加粗**标注。以“+”结尾的方法表示使用 Mask2Former 的解码器进行增强的版本。“†”表示该结果在 ImageNet-22K 数据集上进行了预训练，并采用了 640×640 的更大输入尺寸。

方法	FLOPs ↓	#Params ↓	mIoU ↑
Mask2Former (Swin-T [18])	74G	47M	47.7%
Mask2Former (Swin-S [18])	98G	69M	51.3%
P2T-T+ [21]	56G	31M	48.2%
P2T-S+ [21]	70G	43M	49.6%
P2T-B+ [21]	109G	74M	52.5%
LRFormer-T+ (本文)	53G	31M	**49.4%**
LRFormer-S+ (本文)	70G	48M	**51.3%**
LRFormer-B+ (本文)	94G	80M	**53.7%**
MaskFormer (Swin-B [18])†	195G	102M	53.1%
Mask2Former (Swin-B [18])†	223G	107M	53.9%
Mask DINO (Swin-B [18])†	265G	110M	54.2%
SeMask (Swin-B [18])†	227G	110M	54.4%
LRFormer-L+† (本文)	192G	119M	**55.8%**
MaskFormer (Swin-L [18])†	375G	212M	54.3%
Mask2Former (Swin-L [18])†	403G	215M	56.1%
Mask DINO (Swin-L [18])†	431G	223M	56.6%
SeMask (Swin-L [18])†	426G	223M	56.3%
LRFormer-XL+† (本文)	365G	205M	**58.1%**

显存占用和 FLOPs。 本文的 LRSA 的计算复杂度很低，仅为 $O(C^2+CN)$。本文对 LRFormer 在不同输入尺寸下的效率进行了数值上的分析，并与代表性方法 SegFormer [10] 进行了对比。FLOPs、注意力 FLOPs 以及训练显存的分析结果见 表12。在统计 LRFormer 时，本文还额外计入了上采样操作的计算开销。结果显示，LRFormer-S 在显存占用和 FLOPs 方面均远低于 SegFormer-B2。以 1024 × 1024 的输入为例，LRFormer 中 MHSA 操作的 FLOPs 仅为 0.9 G，而 SegFormer 中自注意力的 FLOPs 高达 54 G，差距显著。本文还观察到，随着输入尺寸的进一步增大，LRFormer 的优势会更加明显，因为在本文的 MHSA 中，增大输入尺寸仅会轻微增加上采样操作的 FLOPs。

4.5 基于查询解码器的进阶 LRFormer

近年来，出现了一些基于查询的框架，如 MaskFormer 系列 MaskFormer [42]、Mask2Former [43]。尽管它们的解码器相比 SegFormer 等直接融合策略要复杂一些，但凭借 Transformer 的优势，在语义分割任务中取得了卓越的性能。如前所述，LRFormer 采用了与先前工作相同的直接融合策略，表明即便是简单的解码方式也能达到最先进的性能。本节探讨 LRFormer 与基于查询的解码器结合的潜力。本文构建了更强大的版本 LRFormer+，即将 LRFormer 编码器与

表 14
不同骨干网络在视觉-语言模型 LISA [90] 上参考分割任务的性能比较。

骨干网络	gIoU (%)	cIoU (%)
ViT-L [16]	36.9	41.1
Swin-L [18]	38.1	43.1
LRFormer-XL (本文)	**40.9**	**45.7**

Mask2Former 解码器配对。本文将其与近期采用查询解码器的方法进行比较，即 Mask2Former [43]、Mask DINO [46] 和 SeMask [89]。由于 Mask DINO 和 SeMask 仅在较大的骨干（如 Swin-L）上提供实现，为了公平比较，本文使用其官方代码在 Swin-B 骨干网络上重新实现了这两种方法。此外，为了更全面的分析，本文还基于近期性能更强的 P2T [21]，结合 Mask2Former 的解码器，构建了 P2T+ 这一对照方法。

本文在 ADE20K 数据集上按照相同的实验设置进行了实验，结果见 表13。LRFormer+ 表现优异，超越了近期的查询式框架 Mask2Former [43] 与 Mask DINO [46]。从表中可以看到，Mask DINO† 在额外增加 42G FLOPs 的计算量的情况下，性能只比 Mask2Former† 提高了 0.2%。SeMask† 的结构更高效，仅增加 4G FLOPs 的计算量便比 Mask2Former† 提升 0.5% 的性能。与采用 Swin-B 骨干的 Mask2Former 相比，升级后的 LRFormer+ 的 mIoU 提高了 1.9%，虽然参数略多，但 FLOPs 大幅减少 31 G，显示了更高的效率。此外，将 Mask2Former 与新颖的 P2T [21] 骨干结合也带来了性能提升，其中 P2T-L 版本达到 52.5% 的 mIoU，比相近 FLOPs 的 Swin-S 版本 Mask2Former 提高 1.2%。然而，LRFormer+ 仍然优于该配置，例如 LRFormer-B+ 相较于增强的 P2T-L+ 进一步提升了 1.2%。

为了提供更直观的分析，本文在图2中可视化了表3和表13的准确率–FLOPs 对比结果。从图中的曲线及数据点可以看出，LRFormer 系列在计算量更低的情况下取得了比其他所有模型（如 Mask2Former [43]、Mask DINO [46] 和 P2T [21]）更高的准确率。

4.6 在视觉-语言模型中的应用

尽管语义分割仍是计算机视觉中的基础任务，研究社区对**推理分割（reasoning segmentation）**展现出日益浓厚的兴趣。这类新兴任务融合了视觉感知与语言理解能力 [90]–[93]。以 LISA [90] 为代表的工作利用诸如 CLIP [91] 和 LLaVA [92] 等大规模视觉–语言模型，依据文本描述对目标进行分割。为展示本文 LRFormer 在语义分割之外的通用性，本文在**参考分割（referring segmentation）**任务上开展实验，以检验该骨干网络是否能够提升视觉–语言模型的性能。

实验设置。 本文以 LISA [90] 搭配 LLaVA-7B-v1 [92] 作为基线模型进行评估。本文仅将 LISA [90] 的视觉骨干网络替换为三种不同选项：ViT-L [16]、Swin-L [18] 以及本文的 LRFormer-XL，同时保持其他组件不变。为确保公平比较，每个骨干均在 COCO 数据集 [94] 上进行预训练，且视觉分支的其余结构保持一致。随后，本文按照官方策略 [90] 训练各方法，并在 ReasonSeg 验证集 [90] 上进行参考分割测试，该任务可根据语言提示分割图像中的特定目标。根据之前的研究工作 [90], [95], [96]，本文也采用 gIoU 与 cIoU 作为评估指标；关于这些指标的更多细节，可参考 [90]。

结果 表14 给出了不同骨干网络的性能对比。本文提出的 LRFormer-XL 骨干网络在两项指标上均显著优于 ViT-L [16] 与 Swin-L [18]。具体而言，LRFormer-XL 在 gIoU 和 cIoU 上分别取得 40.9% 与 45.7% 的成绩，相比 ViT-L [16] 骨干分别提升 4.0% 和 4.6%，相比 Swin-L [18] 骨干分别提升 2.8% 和 2.6%。这些结果验证了本文 LRFormer 在捕获全局上下文的同时，能够保留推理任务所需的细粒度细节信息。无论是在传统语义分割还是指代分割任务中取得的持续性能提升，都凸显了本文架构在多种先进视觉-语言应用中的通用性与潜力。

5 总结

本文通过引入低分辨率自注意力机制（LRSA），提出了一种新的语义分割方法。LRSA 在固定的低分辨率空间内计算自注意力，不受输入图像尺寸的影响，从而显著提高了计算效率。大量实验证明（比如图2），在 ADE20K [28]、COCO-Stuff [29] 和 Cityscapes [30] 数据集上，LRFormer 的性能超越了当前最先进的模型。这验证了 LRSA 能以可忽略的计算开销（即 FLOPs）保持全局感受野的有效性。本研究为 LRSA 的有效性提供了有力证据，并为未来相关研究开辟了新的方向。

致谢。本文工作得到国家自然科学基金（编号：62225604，62176130），中央高校基本科研业务费（南开大学，070-63233089），A*STAR Career Development Fund（编号：C233312006），以及新加坡国家研究基金会 AI Singapore 计划（AISG 奖励编号：AISG2-GC-2023-007）的资助。计算工作部分依托于南开大学超级计算中心完成，另一部分使用了新加坡国家超级计算中心（https://www.nscc.sg）的计算资源。

参考文献

[1] Y.-H. Wu, S.-C. Zhang, Y. Liu, L. Zhang, X. Zhan, D. Zhou, J. Feng, M.-M. Cheng, and L. Zhen, "Low-resolution self-attention for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2025.

[2] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2017.

[3] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 3146–3154.

[4] Z. Zhu, M. Xu, S. Bai, T. Huang, and X. Bai, "Asymmetric non-local neural networks for semantic segmentation," in *Int. Conf. Comput. Vis.*, 2019, pp. 593–602.

[5] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 2881–2890.

[6] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "DenseASPP for semantic segmentation in street scenes," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 3684–3692.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 770–778.

[8] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 1492–1500.

[9] Z. Huang, X. Wang, Y. Wei, L. Huang, H. Shi, W. Liu, and T. S. Huang, "CCNet: Criss-cross attention for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 06, pp. 6896–6908, 2023.

[10] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *Adv. Neural Inform. Process. Syst.*, vol. 34, pp. 12 077–12 090, 2021.

[11] Y. Yuan, R. Fu, L. Huang, W. Lin, C. Zhang, X. Chen, and J. Wang, "Hrformer: High-resolution vision transformer for dense predict," *Adv. Neural Inform. Process. Syst.*, vol. 34, pp. 7281–7293, 2021.

[12] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. C. Loy, D. Lin, and J. Jia, "PSANet: Point-wise spatial attention network for scene parsing," in *Eur. Conf. Comput. Vis.*, 2018, pp. 267–283.

[13] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 7151–7160.

[14] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Learning a discriminative feature network for semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 1857–1866.

[15] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, "Gated-scnn: Gated shape cnns for semantic segmentation," in *Int. Conf. Comput. Vis.*, 2019, pp. 5229–5238.

[16] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Int. Conf. Learn. Represent.*, 2021.

[17] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," *arXiv preprint arXiv:2012.12877*, 2020.

[18] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Int. Conf. Comput. Vis.*, 2021, pp. 10 012–10 022.

[19] X. Dong, J. Bao, D. Chen, W. Zhang, N. Yu, L. Yuan, D. Chen, and B. Guo, "Cswin transformer: A general vision transformer backbone with cross-shaped windows," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 12 124–12 134.

[20] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Int. Conf. Comput. Vis.*, 2021, pp. 568–578.

[21] Y.-H. Wu, Y. Liu, X. Zhan, and M.-M. Cheng, "P2T: Pyramid pooling transformer for scene understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 12 760–12 771, 2023.

[22] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," in *Eur. Conf. Comput. Vis.* Springer, 2020, pp. 173–190.

[23] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Int. Conf. Learn. Represent.*, 2016.

[24] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "PVT v2: Improved baselines with pyramid vision transformer," *Computational Visual Media*, vol. 8, no. 3, pp. 415–424, 2022.

[25] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong *et al.*, "Swin transformer v2: Scaling up capacity and resolution," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 12 009–12 019.

[26] J. Yang, C. Li, P. Zhang, X. Dai, B. Xiao, L. Yuan, and J. Gao, "Focal self-attention for local-global interactions in vision transformers," *arXiv preprint arXiv:2107.00641*, 2021.

[27] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, and C. Feichtenhofer, "Multiscale vision transformers," in *Int. Conf. Comput. Vis.*, 2021, pp. 6824–6835.

[28] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 633–641.

[29] H. Caesar, J. Uijlings, and V. Ferrari, "COCO-Stuff: Thing and stuff classes in context," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 1209–1218.

[30] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 3213–3223.

[31] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 3431–3440.

[32] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[33] H. Ding, X. Jiang, A. Q. Liu, N. M. Thalmann, and G. Wang, "Boundary-aware feature propagation for scene segmentation," in *Int. Conf. Comput. Vis.*, 2019, pp. 6819–6829.

[34] X. Li, X. Li, L. Zhang, G. Cheng, J. Shi, Z. Lin, S. Tan, and Y. Tong, "Improving semantic segmentation via decoupled body and edge supervision," in *Eur. Conf. Comput. Vis.* Springer, 2020, pp. 435–452.

[35] Y. Yuan, J. Xie, X. Chen, and J. Wang, "Segfix: Model-agnostic boundary refinement for segmentation," in *Eur. Conf. Comput. Vis.* Springer, 2020, pp. 489–506.

[36] M. Zhen, J. Wang, L. Zhou, S. Li, T. Shen, J. Shang, T. Fang, and L. Quan, "Joint semantic segmentation and boundary detection using iterative pyramid contexts," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 13 666–13 675.

[37] Y. Yuan, L. Huang, J. Guo, C. Zhang, X. Chen, and J. Wang, "Ocnet: Object context network for scene parsing," *arXiv preprint arXiv:1809.00916*, 2018.

[38] Y.-H. Wu, S.-H. Gao, J. Mei, J. Xu, D.-P. Fan, R.-G. Zhang, and M.-M. Cheng, "JCS: An explainable covid-19 diagnosis system by joint classification and segmentation," *IEEE Trans. Image Process.*, vol. 30, pp. 3113–3126, 2021.

[39] Y. Wang, Y. Li, J. H. Elder, R. Wu, and H. Lu, "Class-conditional domain adaptation for semantic segmentation," *Computational Visual Media*, vol. 10, no. 5, pp. 1013–1030, 2024.

[40] D. Liang, Y. Sun, Y. Du, S. Chen, and S.-J. Huang, "Relative difficulty distillation for semantic segmentation," *Science China Information Sciences*, vol. 67, no. 9, p. 192105, 2024.

[41] Z. Li, W. Wang, E. Xie, Z. Yu, A. Anandkumar, J. M. Alvarez, P. Luo, and T. Lu, "Panoptic segformer: Delving deeper into panoptic segmentation with transformers," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 1280–1289.

[42] B. Cheng, A. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," *Adv. Neural Inform. Process. Syst.*, vol. 34, pp. 17 864–17 875, 2021.

[43] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 1290–1299.

[44] W. Zhang, J. Pang, K. Chen, and C. C. Loy, "K-net: Towards unified image segmentation," *Adv. Neural Inform. Process. Syst.*, vol. 34, pp. 10 326–10 338, 2021.

[45] Q. Yu, H. Wang, S. Qiao, M. Collins, Y. Zhu, H. Adam, A. Yuille, and L.-C. Chen, "K-means mask transformer," in *Eur. Conf. Comput. Vis.* Springer, 2022, pp. 288–307.

[46] F. Li, H. Zhang, H. Xu, S. Liu, L. Zhang, L. M. Ni, and H.-Y. Shum, "Mask DINO: Towards a unified transformer-based framework for object detection and segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023, pp. 3041–3050.

[47] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr *et al.*, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 6881–6890.

[48] J.-h. Shim, H. Yu, K. Kong, and S.-J. Kang, "Feedformer: Revisiting transformer decoder for efficient semantic segmentation," in *AAAI Conf. Artif. Intell.*, vol. 37, no. 2, 2023, pp. 2263–2271.

[49] Y. Fang, W. Wang, B. Xie, Q. Sun, L. Wu, X. Wang, T. Huang, X. Wang, and Y. Cao, "EVA: Exploring the limits of masked visual representation learning at scale," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023, pp. 19 358–19 369.

[50] J. Jain, J. Li, M. T. Chiu, A. Hassani, N. Orlov, and H. Shi, "Oneformer: One transformer to rule universal image segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023, pp. 2989–2998.

[51] P. Wang, S. Wang, J. Lin, S. Bai, X. Zhou, J. Zhou, X. Wang, and C. Zhou, "ONE-PEACE: Exploring one general representation model toward unlimited modalities," *arXiv preprint arXiv:2305.11172*, 2023.

[52] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Adv. Neural Inform. Process. Syst.*, vol. 25, pp. 1097–1105, 2012.

[53] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[54] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 1–9.

[55] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 4700–4708.

[56] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2Net: A new multi-scale backbone architecture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 652–662, 2019.

[57] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Mueller, R. Manmatha *et al.*, "ResNeSt: Split-attention networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 2736–2746.

[58] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 7132–7141.

[59] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 510–519.

[60] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 11 976–11 986.

[61] X. Ding, X. Zhang, J. Han, and G. Ding, "Scaling up your kernels to 31x31: Revisiting large kernel design in CNNs," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 11 963–11 975.

[62] S. Liu, T. Chen, X. Chen, X. Chen, Q. Xiao, B. Wu, M. Pechenizkiy, D. Mocanu, and Z. Wang, "More convnets in the 2020s: Scaling up kernels beyond 51x51 using sparsity," *arXiv preprint arXiv:2207.03620*, 2022.

[63] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, 2020.

[64] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Adv. Neural Inform. Process. Syst.*, 2017, pp. 5998–6008.

[65] D. Zhou, Z. Yu, E. Xie, C. Xiao, A. Anandkumar, J. Feng, and J. M. Alvarez, "Understanding the robustness in vision transformers," in *Int. Conf. Mach. Learn.* PMLR, 2022, pp. 27 378–27 394.

[66] W. Yu, M. Luo, P. Zhou, C. Si, Y. Zhou, X. Wang, J. Feng, and S. Yan, "Metaformer is actually what you need for vision," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 10 819–10 829.

[67] Z. Xia, X. Pan, S. Song, L. E. Li, and G. Huang, "Vision transformer with deformable attention," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 4794–4803.

[68] Y. Liu, Y.-H. Wu, G. Sun, L. Zhang, A. Chhatkuli, and L. Van Gool, "Vision transformers with hierarchical attention," *Machine Intelligence Research*, vol. 21, no. 4, pp. 670–683, 2024.

[69] M.-H. Guo, T.-X. Xu, J.-J. Liu, Z.-N. Liu, P.-T. Jiang, T.-J. Mu, S.-H. Zhang, R. R. Martin, M.-M. Cheng, and S.-M. Hu, "Attention mechanisms in computer vision: A survey," *Computational visual media*, vol. 8, no. 3, pp. 331–368, 2022.

[70] W. Xu, Y. Xu, T. Chang, and Z. Tu, "Co-scale conv-attentional image transformers," in *Int. Conf. Comput. Vis.*, 2021, pp. 9981–9990.

[71] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[72] X. Chu, Z. Tian, B. Zhang, X. Wang, X. Wei, H. Xia, and C. Shen, "Conditional positional encodings for vision transformers," in *Int. Conf. Learn. Represent.*, 2023.

[73] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[74] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 1290–1299.

[75] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Int. Conf. Mach. Learn.* PMLR, 2021, pp. 10 347–10 357.

[76] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Int. Conf. Learn. Represent.*, 2018.

[77] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, and H. Jégou, "Going deeper with image transformers," in *Int. Conf. Comput. Vis.*, 2021, pp. 32–42.

[78] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang, "Vision mamba: Efficient visual representation learning with bidirectional state space model," in *Int. Conf. Mach. Learn.*, 2024, pp. 62 429–62 442.

[79] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Int. Conf. Comput. Vis.*, 2021, pp. 12 179–12 188.

[80] M. Ding, B. Xiao, N. Codella, P. Luo, J. Wang, and L. Yuan, "Davit: Dual attention vision transformers," in *Eur. Conf. Comput. Vis.* Springer, 2022, pp. 74–92.

[81] A. Hatamizadeh, G. Heinrich, H. Yin, A. Tao, J. M. Alvarez, J. Kautz, and P. Molchanov, "FasterViT: Fast vision transformers with hierarchical attention," in *Int. Conf. Learn. Represent.*, 2024.

[82] W. Wang, J. Dai, Z. Chen, Z. Huang, Z. Li, X. Zhu, X. Hu, T. Lu, L. Lu, H. Li *et al.*, "Internimage: Exploring large-scale vision foundation models with deformable convolutions," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023, pp. 14 408–14 419.

[83] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[84] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 761–769.

[85] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," in *Eur. Conf. Comput. Vis.*, 2018, pp. 418–434.

[86] Y. Li, C.-Y. Wu, H. Fan, K. Mangalam, B. Xiong, J. Malik, and C. Feichtenhofer, "MViTv2: Improved multiscale vision transformers for classification and detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 4804–4814.

[87] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, "Designing network design spaces," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 10 428–10 436.

[88] J. Yang, C. Li, and J. Gao, "Focal modulation networks," *arXiv preprint arXiv:2203.11926*, 2022.

[89] J. Jain, A. Singh, N. Orlov, Z. Huang, J. Li, S. Walton, and H. Shi, "Semask: Semantically masked transformers for semantic segmentation," in *International Conference on Computer Vision Workshops*, 2023, pp. 752–761.

[90] X. Lai, Z. Tian, Y. Chen, Y. Li, Y. Yuan, S. Liu, and J. Jia, "LISA: Reasoning segmentation via large language model," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024, pp. 9579–9589.

[91] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *Int. Conf. Mach. Learn.* PmLR, 2021, pp. 8748–8763.

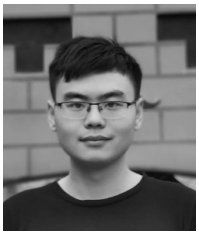
[92] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in *Adv. Neural Inform. Process. Syst.*, 2023, pp. 34 892–34 916.

[93] J. Li, Y. Huang, M. Wu, B. Zhang, X. Ji, and C. Zhang, "CLIP-SP: Vision-language model with adaptive prompting for scene parsing," *Computational Visual Media*, vol. 10, no. 4, pp. 741–752, 2024.

[94] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Eur. Conf. Comput. Vis.* Springer, 2014, pp. 740–755.

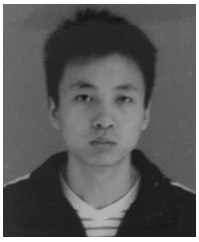
[95] S. Kazemzadeh, V. Ordonez, M. Matten, and T. Berg, "Referitgame: Referring to objects in photographs of natural scenes," in *Proceedings of the 2014 conference on empirical methods in natural language processing*, 2014, pp. 787–798.

[96] J. Mao, J. Huang, A. Toshev, O. Camburu, A. L. Yuille, and K. Murphy, "Generation and comprehension of unambiguous object descriptions," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 11–20.



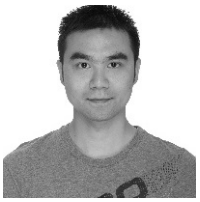
吴宇寰 于 2022 年在南开大学获得博士学位，导师为程明明教授。目前，他是新加坡科技研究局 (A*STAR) 高性能计算研究所 (IHPC) 的研究员。已在 IEEE TPAMI、TIP 以及 CVPR、ICCV 等顶级期刊与会议发表论文 10 余篇。其研究方向涵盖计算机视觉、医学影像和自动驾驶。

张世辰 于 2023 年在南开大学获得计算机科学工学学士学位。目前，他是南开大学媒体计算实验室的博士研究生，师从程明明教授。其研究方向包括目标检测和语义分割。



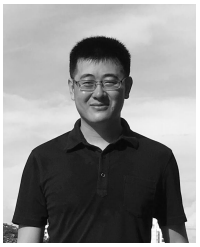
刘云 于 2016 年和 2020 年分别在南开大学获得工学学士和博士学位。之后，他在苏黎世联邦理工学院（ETH Zurich）计算机视觉实验室与 Luc Van Gool 教授合作，担任博士后研究员一年半。目前，他是南开大学教授，研究方向为计算机视觉与机器学习。

张乐 于 2012 年和 2016 年分别在南洋理工大学（Nanyang Technological University, NTU）获得硕士和博士学位。目前，他是电子科技大学教授。曾担任 AAAI、IJCAI 等多个国际会议的程序委员会成员，并受邀担任《Pattern Recognition》和《Neurocomputing》期刊的客座编辑。其研究方向包括深度学习和计算机视觉。



占新 于 2010 年和 2015 年分别在中国科学技术大学（USTC）获得学士和博士学位。2015 年至 2023 年间，他在阿里巴巴集团担任算法专家，专注于大规模人工智能应用。目前，他是有鹿机器人科技的算法负责人，主导下一代机器人系统的研究工作。他的研究方向包括端到端自动驾驶、具身智能，以及多模态视觉-语言-动作（VLA）模型。

周大权 博士毕业于新加坡国立大学（NUS），导师为冯佳时教授（Prof. Jiashi Feng）。现为北京大学助理教授。其研究方向包括深度学习、神经网络压缩、神经网络结构设计以及自动机器学习（AutoML）。



冯佳时 于 2014 年在新加坡国立大学（NUS）获得博士学位。现任字节跳动（ByteDance）研究负责人。在加入字节跳动之前，他曾任新加坡国立大学电气与计算机工程系助理教授。他的研究领域包括深度学习及其在计算机视觉中的应用。他曾获得 ACM MM 2012 最佳技术演示奖、ICCV 2015 TASK-CV 最佳论文奖以及 ACM MM 2018 最佳学生论文奖。

程明明 于 2012 年在清华大学获得博士学位，随后在牛津大学与 Philip Torr 教授合作，从事两年博士后研究工作。目前，他是南开大学教授，媒体计算实验室(Media Computing Lab)的领导。他的研究方向包括计算机图形学、计算机视觉和图像处理。曾获多项研究奖项，包括 ACM China Rising Star Award、IBM Global SUR Award 和 CCF-Intel 青年教师科研奖。他现担任 IEEE TPAMI 和 IEEE TIP 的编委。

甄亮利 于 2018 年在四川大学获得博士学位。现任新加坡科技研究局（A*STAR）高性能计算研究所（IHPC）高级科学家兼课题组负责人。他的研究兴趣包括机器学习与优化。他曾主持或联合主持多个国家级重大项目，包括新加坡航空航天计划（Singapore Aerospace Programme）、AI Singapore 鲁棒人工智能重大挑战项目（Robust AI Grand Challenge）以及 AI Singapore 技术挑战（Technology Challenge）。