arXiv:2304.02942v2 [cs.CV] 9 Aug 2023

InterFormer 实时交互式图像分割

You Huang [1], Hao Yang [1], Ke Sun [1], Shengchuan Zhang [1*], Liujuan Cao[1], Guannan Jiang [2], Rongrong Ji[1] 中华人民共和国教育部多媒体可信感知与高效计算重点实验室，厦门大学[2] 宁德时代新能源科技股份有限公司（宁德时代）智能制造部门

摘要

交互式图像分割使标注人员能够高效地完成分割任务的像素级标注。然而，由于以下两个问题，现有的交互式分割处理流程中的交互式模型存在计算效率低下的问题。首先，标注人员后续的点击依赖于模型对其先前点击的反馈。这种串行交互无法利用模型的并行能力。其次，在每个交互步骤中，模型都要处理不变的图像以及稀疏且可变的点击，导致处理过程高度重复且冗余。为实现高效计算，我们提出了一种名为 *InterFormer* 的方法，采用新的处理流程来解决这些问题。*InterFormer* 从现有流程中提取并预处理计算耗时的部分，即图像处理。具体而言，*InterFormer* 在高性能设备上使用大型视觉变换器（*ViT*）并行预处理图像，然后使用名为交互式多头自注意力（*I-MSA*）的轻量级模块进行交互式分割。此外，将 *I-MSA* 模块部署在低功耗设备上，拓展了交互式分割的实际应用。*I-MSA* 模块利用预处理后的特征，高效地实时响应标注人员的输入。在多个数据集上的实验表明，*InterFormer* 具有良好的有效性；与以往的交互式分割模型相比，它在计算效率和分割质量方面均表现更优，并能够在仅使用 *CPU* 的设备上实现实时、高质量的交互式分割。代码发布于 *https://github.com/YouHuang67/InterFormer*。

1. 引言

在海量数据的推动下 [46], ，深度网络在各种计算机视觉任务中取得了令人瞩目的表现

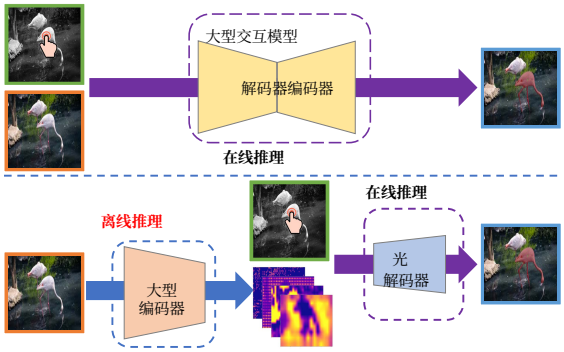


图 1. 现有交互式处理流程与所提出处理流程的示意图。现有处理流程（上）在每次交互期间都向交互式模型提供图像和标注者点击，而所提出的处理流程（下）采用大型编码器对图像进行预处理，然后在交互期间向轻量级解码器提供固定的编码特征和即时点击。

[6, 8, 36]。准确标注数据的可用性对于深度网络的成功至关重要。然而，手动标注的过程既耗时又耗费资源。因此，交互式图像分割已成为标注大规模图像数据集不可或缺的工具 [5]。这类技术旨在通过有限的标注者交互实现高质量的像素级标注，包括涂鸦 [4], 、边界框 [43, 50], 、多边形 [1, 35], 、点击 [2, 41, 10, 33, 38],以及它们的一些组合 [57]。其中，基于点击的方法因其简单易用而最为普遍，本文将重点讨论这类方法。

近期基于点击的交互式分割研究主要集中于各种细化模块 [10, 33] ，这些模块融-合了复杂的工程优化。然而，这类细化技巧仍然需要在交互过程中从早期交互中获得分割良好的结-果。在低功耗设备上部署计算密集型模型时，生成此类结果仍面临挑战。例如，通过众包平台 [3]使用交互式分-割模型颇具挑战。先前-

*通讯作者

诸如 FocalClick 这样的工作 [10] 通过使用轻量级模型和对输入进行下采样来缓解这一问题，但这种策略以牺牲分割质量为代价。因此，仍然需要适用于各种设备的计算友好型交互式分割方法。

交互式分割效率低下源于两个根本原因。首先，标注人员每次点击都对应一次模型推理，而下一次点击的位置取决于前一次推理的结果。这种串行交互每次推理只能处理一个样本，无法利用 GPU 的并行计算能力。其次，在单张图像的整个标注过程中，模型输入始终非常相似，只有稀疏点击会发生变化。因此，每次推理提取的特征几乎相同，造成了大量计算冗余。这两个问题导致计算效率低下。

本文提出了一种名为 InterFormer 的方法，以提高计算效率。在交互过程开始之前，InterFormer 会在高性能设备上使用大型模型例如视觉变换器（ViT）[12]，从待标注图像中提取高质量特征。此过程在线下进行，无需满足实时性能要求。随后，InterFormer 只需借助轻量级模块，即可在低功耗设备上执行交互式分割，其输入为大型模型预处理后的特征和标注人员的点击。

继之前在编码标注者点击方面的工作之后 [27]，我们尝试通过类似特征金字塔网络的卷积网络来实现轻量级模块 [31]，以将点击与预处理后的特征进行融合。然而，该模块无法有效利用预处理后的特征，并产生了不理想的分割结果（见第4.3节）。因此，我们提出了交互式多头自注意力（I-MSA），这是一种高效的交互式分割模块，其灵感来自视觉变换器变体近期取得的成功 [39, 49, 51]。I-MSA 的计算复杂度极低，并针对充分利用大型模型生成的预处理特征进行了优化。此外，我们采用放大策略 [44]，并对I-MSA 的深层块进行轻微修改，使其聚焦于图像中潜在物体周围的有效区域。这些修改后的块使 I-MSA 的推理速度显著提升，同时性能仅略有下降。

如图2所示，所提出的 InterFormer 优于以往的交互式分割模型，并以相近的计算成本取得了最先进的性能。为公平比较，计算速度的测量将 ViT 在同一设备上提取特征的过程也纳入考量。此外，基于离线预处理的特征，InterFormer 能够在仅使用 CPU 的设备上实现实时、高质量的交互式分割。我们在 GrabCut [43], Berkeley [26], SBD [20], 和 DAVIS [42]

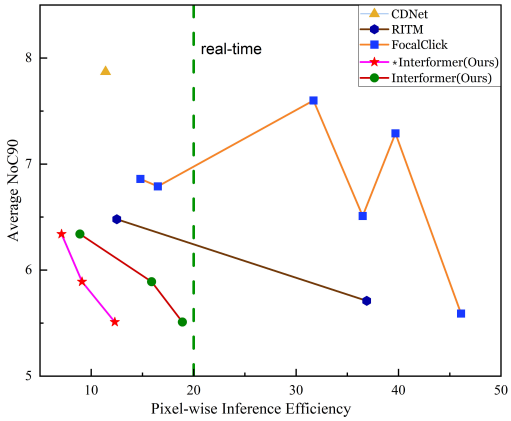


图 2. 在 SBD数据集 [20] 上进行交互式分割的实验结果。平均 NoC90 表示达到 0.9 IoU 所需的平均点击次数，PIE（逐像素推理效率）衡量模型对可能较大或较小的物体进行逐像素分割的效率。我们提出的 InterFormer（∗ 表示未考虑预处理时的 PIE 测量值）展现出比近期提出的方法更快的推理速度和更好的分割效果。

数据集上进行了广泛的实验，以验证 InterFormer 的有效性。我们的贡献总结如下：

- 我们提出了一种名为 InterFormer 的方法，它采用全新的处理流程，将交互过程分为两个阶段，从而充分利用成熟的大规模模型并加速交互。
- 我们提出了一种名为 I-MSA 的交互注意力模块，利用预处理后的特征，在仅使用 CPU 的设备上实现高质量的实时交互式分割。
- 在计算效率和分割质量方面，InterFormer 显著优于以往方法。

2. 相关工作

2.1. 交互式分割

在深度网络出现之前，基于图的方 [7, 16, 43] 法在交互式图像分割研究中十分普遍。这些方法将图像表示为图，并使用优化方法，根据标注者输入求解图割问题。然而，这类方法使用的低层特征仅适用于常规的图像分割场景。DIOS [55]率先将深度网络引入交互式分割，并提出一种将点击转换为距离图、再与图像拼接的策略。DIOS 还为基于点击的方法制定了训练/测试处理流程。此后，人们提出了多种方法来改进基于点击的交互式分割，包括 DEXTR [41], FCA-Net [34], BRS [25] 和

f-BRS [44]。这些方法着眼于交互式分割的不同方面，以提升效率，例如物体周围的四个极值点 [41], 首次点击 [34],以及推理优化 [25, 44]。近期，RITM [27]被提出，它将先前的分割结果纳入模型输入。其他近期的基于点击的方法 [10, 33]专注于局部细化，并将先前的处理流程分解为粗分割和细化两个阶段，分别由轻量级模型实现。此外，PseudoClick [38]通过额外模块及相应损失来模拟标注者点击。本文提出了一种有别于现有方法的新处理流程。我们的处理流程使用大型模型对图像进行离线预处理，并使用轻量级模型执行交互式分割。

2.2. 视觉 Transformer

Transformer 架构最初是为机器翻译 [48], 提出的，旨在建模全局令牌之间的依赖关系。随后，视觉变换器（ViT） [12] 将这种建模方法应用于计算机视觉任务，通过连接分布在整个图像周围的图像块来实现这一点。这启发了一系列基于 ViT 的工作 [47, 56, 49, 14, 39, 51, 24, 23] ，并成功解决了各种计算机视觉任务。之后，ViT 进一步被发展用于图像分割，例如 SETR [58], 、Segmenter [45]和 SegFormer [53]。最近，FocalClick [10]采用SegFormer 进行交互式分割，而 SimpleClick [37] 则将经过 MAE 预训练的 ViT [21] 引入交互式分割。在本文中，我们遵循 SimpleClick [37]的方法，使用大型纯 ViT 作为编码器进行特征编码。该过程在线下完成，无需满足实时性能要求，并生成了高质量的预处理特征。

3. 方法

我们提出了 InterFormer，它遵循计算高效的处理流程，以解决现有处理流程的低效问题。第 3.1节介绍了新的处理流程。第 3.2节描述了所提出的 I-MSA 模块，用于实现该处理流程。第3.3节介绍了 InterFormer的放大策略。第3.4节总结了一种简化的训练设置。

3.1. 高效计算处理流程

所提出的处理流程将基于大型模型的交互分解为三个不同的过程：特征编码、点击嵌入和特征解码（如图3所示）。特征编码使大型模型能够离线预处理图像并提取可重复使用的特征，从而无需满足实时性能要求。在交互过程中，该处理流程会转换标注者点击转换为嵌入，然后向解码器提供预处理后的特征和嵌入，以生成分割结果。

特征编码。 受 Sim- pleClick 工作的启发 [37], 我们采用大型模型进行特征编码，例如广泛使用的 MAE 预训练视觉变换器 (ViT) [21]。此外，遵循 ViT- Det 中的方法 [28], 我们使用简单的特征金字塔网络 [31]从基础 ViT 最后一层中的单尺度嵌入块生成多尺度特征。例如，ViT-Base (ViT- B) [21] 将尺寸为 $H \times W$ 的输入图像划分为一系列 16×16 个块，然后将其投影为 C_0维向量。ViT 块对这些定长向量执行多头自注意力，生成$\frac{H}{16} \times \frac{W}{16}$ C_0维向量。我们使用简单的特征金字塔网络将这些向量转换为特征图 $\boldsymbol{F}_i$, $1 \leq i \leq 4$，通道数为 $2^{i-1}C_1, 1 \leq i \leq 4$，尺寸分别为 $\frac{H}{4} \times \frac{H}{4}$、$\frac{H}{8} \times \frac{H}{8}$、$\frac{H}{16} \times \frac{H}{16}$ 和 $\frac{H}{32} \times \frac{H}{32}$，通过多层卷积和池化实现。该模块既能提取多尺度信息，也能降低后续解码的计算开销。

点击嵌入。 交互过程中，标注人员会在模型预测错误的区域中单击一次；正向点击和负向点击分别对应假阴性区域和假阳性区域。以往的方法 [27, 10] 通过在图像通道中添加两张点击周围包含圆盘的图来编码此类点击。这种策略会轻微修改后续的点击图。RITM 流程 [27] 进一步通过将先前的分割结果加入图像通道来改进分割。我们采用这些策略，将点击图与先前的结果融合，而不是添加更多通道。

我们维护一个大小为 $H \times W$ 的掩码 $\boldsymbol{M}_{\text{ref}}$，并在每次交互时更新它。根据前景/背景置信度，将 $\boldsymbol{M}_{\text{ref}}$ 中的每个像素归类为 $[D_{\text{fg}}]$、$[P_{\text{fg}}]$、$[U]$、$[P_{\text{bg}}]$ 或 $[D_{\text{bg}}]$，其中“P”表示“可能”，“D”表示“确定”，“bg”指背景，“fg”指前景，“U”表示“未知”。最初，$\boldsymbol{M}_{\text{ref}}$ 中的所有像素均设为 $[U]$。在交互过程中，每次新点击都会在 $\boldsymbol{M}_{\text{ref}}$ 上引入一个半径为五个像素的圆盘。根据点击是正向还是负向，将该圆盘内的像素标记为 $[D_{\text{fg}}]$ 或 $[D_{\text{bg}}]$。纳入点击后，模型会预测分割结果，将识别为前景或背景的像素分别标记为 $[P_{\text{fg}}]$ 或 $[P_{\text{bg}}]$。点击圆盘内的标签保持不变。根据 $\boldsymbol{M}_{\text{ref}}$ 的当前状态和分割预测，为这些圆盘之外的像素分配 $[P_{\text{fg}}]$、$[P_{\text{bg}}]$ 或 $[U]$ 中的一个标签。分配遵循以下规则：$[U] + [P_{\text{fg}}] \rightarrow [P_{\text{fg}}]$、$[U] + [P_{\text{bg}}] \rightarrow [P_{\text{bg}}]$ 和 $[P_{\text{fg}}] + [P_{\text{bg}}] \rightarrow [U]$。

在模型推理过程中，M_{ref}中的每个标签类别都对应一个特定的可学习 C_1维向量。

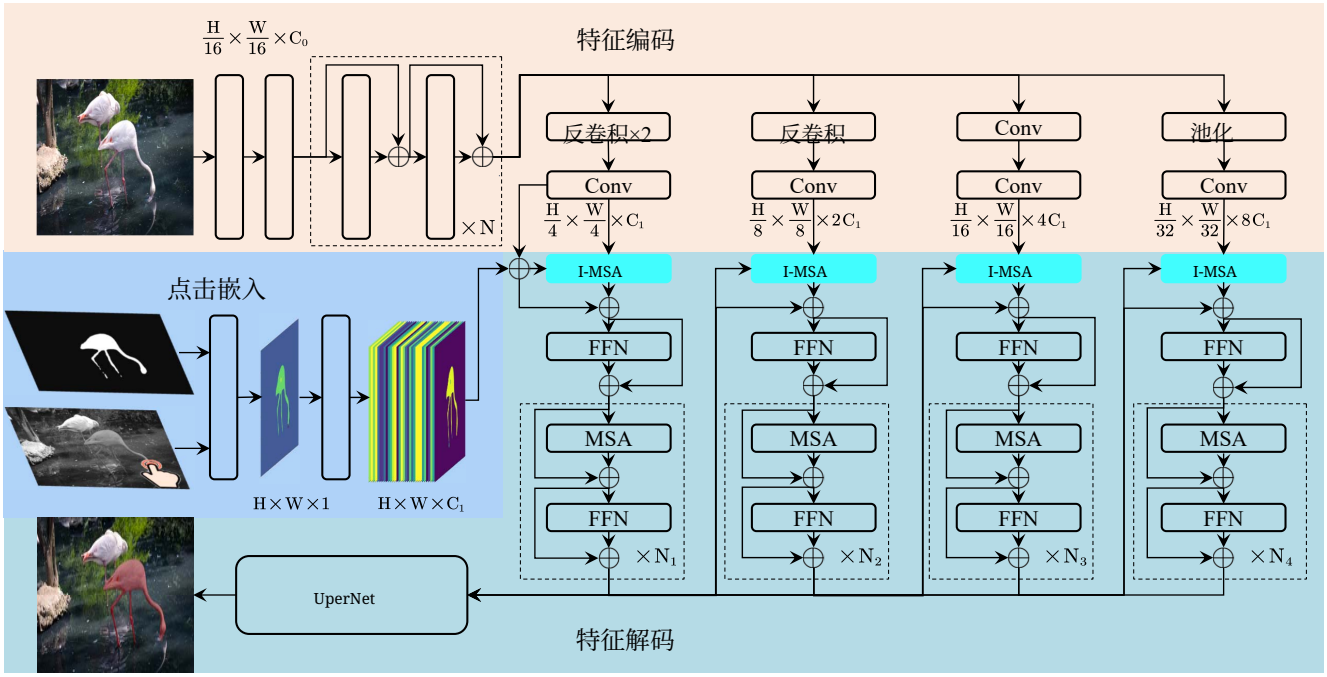


图3. 我们提出的 InterFormer 概览。该流程将特征提取与交互分离。首先，特征编码模块从输入图像中提取多尺度特征。接下来，点击嵌入模块生成点击图，并将其与图像特征融合。最后，特征解码模块整合多尺度特征，并将其解码为最终分割结果。

此映射会将 M_{ref} 转换为点击嵌入 $\boldsymbol{E}_c \in \mathcal{R}^{H\times W\times C_1}$。为了生成包含点击信息的特征 $\boldsymbol{F}_c$，我们随后调整 $\boldsymbol{E}_c$ 的尺寸，并将其与来自特征金字塔网络模块的特征 $\boldsymbol{F}_1$ 结合：$\boldsymbol{F}_c = \text{resize}(\boldsymbol{E}_c) + \boldsymbol{F}_1$。

特征解码。我们进一步融合编码后的特征$\boldsymbol{F}_i$，$1 \le i \le 4$与点击相关的特征 $\boldsymbol{F}_c$，使用所提出的 I-MSA 模块（在第3.2节介绍）。随后，使用广泛应用的 UperNet [52]对融合结果进行解码，生成最终的分割结果。

3.2. 交互式多头自注意力

以往工作 [27, 10, 33, 38] 采用一种简单的方法处理仅经过编码的点击图：将其扩展为图像的附加通道。然而，要有效地将稀疏的点击信息与图像特征融合，这种方法需要堆叠大量卷积层 [27, 33, 38] 或自注意力模块 [10]，导致计算复杂度较高。在我们的处理流程中，由于解码器架构较浅（见第4.3节），这种方法会导致结果退化。因此，我们提出 I-MSA，以高效地整合点击信息。

交互式自注意力。随着 Transformer的兴起，人们对自注意力机制的关注日益增加，该机制利用图像块嵌入的相似性来筛选出对任务至关重要的图像块 [12]。交互式分割天然适用于这一机制，因为它可-

标注者的点击提供感兴趣区域，模型可以利用这些特征，根据它们的相似性识别感兴趣对象的组成部分。这启发我们将自注意力重新表述为交互式注意力。具体而言，我们将涉及点击的特征 $\boldsymbol{F}_c$ 和预处理后的特征 $\boldsymbol{F}_i, 1 \le i \le 4$ 分别重新定位为自注意力中的查询 $\boldsymbol{Q}$ 和键 $\boldsymbol{K}$，其中值 $\boldsymbol{V}$ 与键共享相同的特征。为方便起见，我们将传统的 $(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})$ [12] 重新表述为一个函数

$$(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})(\boldsymbol{A}, \boldsymbol{B}) = (\boldsymbol{A}\boldsymbol{W}^q, \boldsymbol{B}\boldsymbol{W}^k, \boldsymbol{B}\boldsymbol{W}^v), \quad (1)$$

给定输入 $\boldsymbol{A}$ 和 $\boldsymbol{B}$。接着，我们将常规多头自注意力模块定义为 MSA$(\boldsymbol{A}) = (\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})(\boldsymbol{A}, \boldsymbol{A})$，将提出的 I-MSA 模块定义为 I-MSA$(\boldsymbol{A}, \boldsymbol{B}) = (\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})(\boldsymbol{A}, \boldsymbol{B})$。元组 $(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})$ 通过 softmax 运算用于常规注意力计算。为简洁起见，我们不详细说明这一操作，假设注意力计算是所介绍模块固有的。

如图所示3，我们采用近期视觉变换器变体的层次化架构 [39, 49] 通过从低层特征开始构建层次化表示$\boldsymbol{F}_1$。我们对 $\boldsymbol{F}_1$ 和 $\boldsymbol{F}_c$ 执行交互注意力，以获得 $\boldsymbol{H}_1^1$ = I-MSA$(\boldsymbol{F}_c, \boldsymbol{F}_1)$。然后，我们迭代地对 $\boldsymbol{H}_1^1$ 执行自注意力。换言之，我们计算$\boldsymbol{H}_1^i$ = 多头自注意力$(\boldsymbol{H}_1^{i-1})$，对于每个 $2 \le i \le N_1 + 1$，其中 N_1是第一阶段的深度。随后，我们采用

上的交互式注意力 $\boldsymbol{F}_2$，$\boldsymbol{H}_1^{N_1+1}$ 以启动第二阶段，而不是重新使用普通点击相关特征 $\boldsymbol{F}_c$。此外，在启动第二阶段之前，我们采用了分层 Transformer 中使用的补丁合并操作 [39]于 $\boldsymbol{H}_1^{N_1+1}$。随后与$\boldsymbol{F}_3$和 $\boldsymbol{F}_4$分别对应的两个阶段遵循相同的流程。最后，我们获得特征 $\boldsymbol{H}_i^{N_i+1}$，$1 \leq i \leq 4$，然后将其输入 UperNet 解码器。

基于池化的自注意力。我们通过采用 P2T 的池化策略 [51]，进一步提升了交互式自注意力的效率。该技术有助于缓解自注意力的二次计算复杂度。具体而言，P2T 使用一系列具有不同池化比例的平均池化层，对输入 $\boldsymbol{B}$的 $(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})(\boldsymbol{A}, \boldsymbol{B})$ 进行预处理，即。

$$\begin{aligned} \boldsymbol{P}_1 &= \text{AvgPool}_1(\boldsymbol{B}), \\ \boldsymbol{P}_2 &= \text{AvgPool}_2(\boldsymbol{B}), \\ &\cdots, \\ \boldsymbol{P}_n &= \text{AvgPool}_n(\boldsymbol{B}). \end{aligned} \tag{2}$$

得到的金字塔特征 $\boldsymbol{P}_i, 1 \leq i \leq n$随后经过深度卷积处理、展平并连接，以生成更短的特征序列。最后，这些池化特征被输入到 $(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})(\boldsymbol{A}, \cdot)$ 中，其中 $\boldsymbol{A}$保持不变。

3.3. 放大策略

[44] 中提出的放大策略涉及对原始图像中潜在目标周围的区域进行裁剪和调整大小，使其达到分割模型所需的足够尺寸。然而，由于特征提取已在离线预处理阶段完成，所提出的处理流程与放大策略 [44]不兼容，因此无法按要求对目标周围的裁剪区域进行预处理。因此，我们对 I-MSA 模块进行了细微修改，以便在特征解码中融入这一策略。

具体而言，放大策略是一种在图像分割任务中选择感兴趣区域（RoI）的技术，它通过边界框和标注者点击来确定前景。为确保上下文充分，将 RoI 放大 1.4 倍。随后，将 RoI 坐标映射到特征图 $\boldsymbol{F}_i$ 的坐标系中，方法是将坐标除以特征的步幅（例如4、8、16 或 32）。接着裁剪相关特征，并将其输入 I-MSA 模块。由于特征不连续，我们会稍微扩大 RoI，使其尺寸可被最大步幅（即32）整除，从而无需采用 RoIAlign [22]等插值策略即可裁剪特征。此外，我们只将裁剪后的特征输入各阶段的深层块，并将变换后的裁剪特征注入完整特征中。

3.4. 训练

点击模拟。我们简化了 RITM [27]提出的点击模拟方法，以训练 InterFormer 模型。原方法通过在真实掩码内部或外部随机采样点击来模拟交互过程，随后根据分割结果迭代生成点击。然而，我们发现，去除初始随机点击采样、仅进行迭代模拟，就足以满足需求。为平衡计算成本与模拟次数的需求，我们采用指数衰减概率对模拟次数进行采样。我们的方法在计算速度上与原始模拟过程相当，且无需专门设计模拟策略。

训练监督。我们采用 RITM [27]提出的归一化焦点损失（NFL），该损失已被证明能够使交互式分割模型的训练具有更好的收敛性，并且我们进一步通过实验证明了 NFL 在我们的模型上的有效性。

4. 实验

在第 4.1节中，我们概述所提出的 InterFormer 的基本设置以及训练/测试细节。在第4.2节中，我们比较 InterFormer 与其他现有方法在各种基准数据集上的性能，包括 GrabCut [43], Berkeley、 [26], SBD [20], 和 DAVIS [42]数据集。我们在第4.3 节中报告消融研究结果，以分析 InterFormer 的不同组件所产生的影响。最后，在第4.4节中，我们展示 InterFormer 的定性结果，彰显其在交互式分割方面的有效性。

4.1. 实验设置

模型	InterFormer-Light	InterFormer-Tiny
骨干网络	视觉变换器-Base	ViT-Large
N_1, N_2, N_3, N_4	0, 0, 1, 0	1, 1, 5, 2

表 1. InterFormers 的配置。

模型系列。我们提出了两种 InterFormer 模型配置，如表1所示，可部署于仅使用 CPU 的设备上。两种配置均采用轻量级架构，I-MSA 的四个阶段分别使用 32、64、128 和 256 个通道。此外，UperNet 模型使用 64 个通道。为进一步降低计算复杂度，我们采用放大策略，但由于 InterFormer-Tiny 的架构规模较大，该策略仅用于这一配置。具体而言，我们在每个阶段对深层块应用放大策略，分别从第2个、2个、3个和第2个块开始

方法	训练数据	PIE（1e-7）	SPC（尺寸）	GrabCut [43]	伯克利 [26]	SBD [20]		DAVIS [42]
				NoC 90	NoC 90	NoC 85	NoC 90	NoC 90
图割 [7]	/	–	–	10.00	14.22	13.6	15.96	17.41
测地线抠图 [17]	/	–	–	14.57	15.96	15.36	17.60	19.50
随机游走 [16]	/	–	–	13.77	14.02	12.22	15.04	18.31
欧几里得星凸性 [17]	/	–	–	9.20	12.11	12.21	14.86	17.70
测地星凸性 [17]	/	–	–	9.12	12.57	12.69	15.31	17.52
不含 GC 的 DOS [55]	增强型 VOC [13, 20]	–	–	12.59	–	14.30	16.79	17.11
DOS 与 GC [55]	增强型 VOC [13, 20]	–	–	6.08	–	9.22	12.80	12.58
RIS-Net [30]	增强型 VOC [13, 20]	–	–	5.00	–	6.03	–	–
CM 指导 [40]	增强型 VOC [13, 20]	–	–	3.58	5.60	–	–	–
FCANet（SIS） [34]	增强型 VOC [13, 20]	–	–	2.14	4.19	–	–	7.90
潜在多样性 [29]	SBD [20]	–	–	4.79	–	7.41	10.78	9.57
BRS [25]	SBD [20]	–	–	3.60	5.08	6.59	9.78	8.24
f-BRS-B-resnet50 [44]	SBD [20]	–	–	2.98	4.34	5.06	8.08	7.81
CDNet-resnet50 [9]	SBD [20]	11.4	0.30 (512)	2.64	3.69	4.37	7.87	6.66
RITM-HRNet18 [27]	SBD [20]	–	–	2.04	3.22	3.39	5.43	6.71
FocalClick-HRNet18s-S2 [10]	SBD [20]	16.5	0.11 (256)	2.06	3.14	4.30	6.52	6.48
FocalClick-SegFormerB0-S2 [10]	SBD [20]	14.8	0.10 (256)	1.90	3.14	4.34	6.51	7.06
FocusCut-ResNet-50 [33]	SBD [20]	–	–	1.78	3.44	3.62	5.66	6.38
FocusCut-ResNet-101 [33]	SBD [20]	–	–	1.64	3.01	3.40	5.31	6.22
PseudoClick-HRNet18 [38]	SBD [20]	–	–	2.04	3.23	–	5.40	6.57
PseudoClick-HRNet32 [38]	SBD [20]	–	–	1.84	2.98	–	5.61	6.16
99%AccuracyNet [15]	合成 [11, 20, 32, 54]	–	–	1.80	3.04	3.90	–	–
f-BRS-B-HRNet32 [44]	COCO [32]+LVIS [18]	–	–	1.69	2.44	4.37	7.26	6.50
RITM-HRNet18s [27]	COCO [32]+LVIS [18]	12.5	0.20（400）	1.68	2.60	4.04	6.48	5.98
RITM-HRNet32 [27]	COCO [32]+LVIS [18]	36.9	0.59（400）	1.56	2.10	3.59	5.71	5.34
EdgeFlow-HRNet18 [19]	COCO [32]+LVIS [18]	–	–	1.72	2.40	–	–	5.77
FocalClick-HRNet18s-S1 [10]	COCO [32]+LVIS [18]	39.7	0.07（128）	1.82	2.89	4.74	7.29	6.56
FocalClick-HRNet18s-S2 [10]	COCO [32]+LVIS [18]	16.5	0.11 (256)	1.62	2.66	4.43	6.79	5.25
Focal点击-HRNet32-S2 [10]	COCO [32]+LVIS [18]	36.5	0.24（256）	1.80	2.36	4.24	6.51	5.39
FocalClick-SegFormerB0-S1 [10]	COCO [32]+LVIS [18]	31.7	0.05（128）	1.86	3.29	4.98	7.60	7.42
FocalClick-SegFormerB0-S2 [10]	COCO [32]+LVIS [18]	14.8	0.10 (256)	1.66	2.27	4.56	6.86	5.49
FocalClick-SegFormerB3-S2 [10]	COCO [32]+LVIS [18]	46.1	0.30 (256)	1.50	**1.92**	3.53	5.59	**4.90**
PseudoClick-HRNet32 [38]	COCO [32]+LVIS [18]	–	–	1.50	2.08	–	5.54	5.11
InterFormer-Light	COCO [32]+LVIS [18]	8.9	0.23 (512)	1.50	3.14	3.78	6.34	6.19
* InterFormer-轻量版		7.1	0.19 (512)					
InterFormer-Tiny + 放大	COCO [32]+LVIS [18]	15.9	0.42（512）	1.40	2.78	3.56	5.89	5.52
* InterFormer-Tiny +放大		9.1	0.24 (512)					
InterFormer-Tiny	COCO [32]+LVIS [18]	18.9	0.50 (512)	**1.36**	2.53	**3.25**	**5.51**	5.21
* InterFormer-Tiny		12.3	0.32 (512)					
SimpleClick-视觉变换器-B [37]	COCO [32]+LVIS [18]	75.4	1.51 (448)	1.48	1.97	3.43	5.62	5.06
SimpleClick-ViT-L [37]	COCO [32]+LVIS [18]	165.7	3.33 (448)	1.40	1.89	2.95	4.89	4.81
SimpleClick-ViT-H [37]	COCO [32]+LVIS [18]	386.8	7.76 (448)	1.50	1.75	2.85	4.70	4.78

表 2. InterFormer 在 GrabCut、Berkeley、SBD 和 DAVIS 数据集上的评估结果。InferFormer 的 SPC 和 PIE 是通过对 20 次点击的推理时间取平均值测得的，其中包含图像预处理。 $*$ 表示 SPC 和 PIE 是在交互过程中评估的，不考虑图像预处理。这些结果表明，InterFormer 有望提升交互式分割的分割性能和效率。

在四个阶段中，InterFormer 的 I-MSA 较浅层的块无需采用放大策略即可处理全局特征。

训练策略。 我们的模型使用 COCO [32] 和 LVIS [18], , 遵循先前的工作 [10, 38]进行训练。我们采用了广泛使用的数据增强技术，包括在 0.5 到 2.0 之间随机缩放、随机翻转、随机裁剪和颜色抖动。每张增强后的图像都会填充至最终大小 512×512 像素。由于我们的视觉变换器骨干网络是在 224×224 像素图像上通过 MAE 预训练的 [21],，我们会调整预训练的绝对位置嵌入大小，使其与图像大小相匹配。在训练过程中，我们的模型最多可以执行 20 次交互前操作，将点击模拟为模型输入。我们使用 $\gamma = 0.6$ 的概率衰减比例来采样模拟次数。在消融研究中，我们的模型使用批量大小 16 训练 40k 次迭代，在主要实验中训练 320k 次迭代。我们使用 AdamW 优化器，学习率为 1×10^{-4}，并采用多项式衰减策略；视觉变换器-Base 的层衰减率设为 0。65，ViT-Large 的层衰减率设为 0。75。训练过程中我们不采用任何放大策略。

评估策略。 每张评估图像都会填充至 InterFormer 内部特征最大步幅的倍数（即 32）。视觉变换器的位置嵌入会根据输入图像尺寸动态调整

以匹配填充图像。与先前的工作 [27, 10], 一致，我们采用 InterFormer 迭代执行交互式分割，最大点击次数设为20。每次点击都放置在错误区域的中心。此外，我们采用放大策略，以更快的模型推理速度评估 InterFormer-Tiny 的性能。

评估指标。我们报告每种方法的 NoC IoU（点击次数），该指标表示达到目标 IoU 所需的平均点击次数。此外，我们重点评估 InterFormer 在仅使用 CPU 的设备上的速度，以深入了解交互式分割模型的实际性能。

为评估 InterFormer 的速度，我们采用广泛使用的指标：每次点击秒数（SPC）。然而，每种交互式分割模型都有各自的缩放策略，导致 SPC 评估结果不一致。例如，Fo- calClick [10] 会将图像下采样至 256×256 尺寸，以获得粗略的分割结果，然后再通过额外模块进行细化。为解决评估不一致的问题，我们提出了一种名为像素级推理效率（PIE）的新指标，用于衡量模型对可能较大或较小物体进行像素级分割的效率。我们通过计算 SPC 的像素级平均值来评估 PIE。

模型	训练	NoC85/NoC90
视觉变换器-B-特征金字塔网络	RITM	7.53/10.70
视觉变换器-B-I-MSA-Light	RITM	5.38/8.27
视觉变换器-B-I-MSA-Light	简单	5.35/8.34
VIT-L-I-MSA-Light	简单	4.80/7.71

表 3. InterFormer 的消融研究。我们发现，与 ViT-Base 相比，ViT-Large 等更大的骨干网络可显著提升模型性能，而交互模块在提升交互式分割模型性能方面发挥着更为关键的作用。相比之下，训练策略的选择对性能的影响较小。

4.2. 主要结果

在现有基准上的性能。我们展示了 InterFormer 的结果，如表 4.2所示。为便于比较，我们根据训练数据或 PIE 将先前的方法分为不同组，因为更大的训练数据集和计算量通常会带来更好的性能。在允许仅使用 CPU 的设备进行实时交互的方法中，我们提出的 InterFormer-Tiny 在最大的验证数据集 SBD [20], 上，在 NoC85 和 NoC90 两项指标上均达到了最先进的性能，该数据集包含 6671 张图像。此外，InterFormer 在其他数据集上也展现出具有竞争力的性能。不过，值得注意的是，FocalClick [10] 的表现优于 InterFormer 在 DAVIS 数据集上，由于其计算量显著更大（以 PIE 衡量），表现不佳。此外，InterFormer 展现出更高的效率，整体 PIE 通常最低，表明无论图像和物体大小如何变化，它在仅使用 CPU 的设备上进行交互式分割时都具有很高的效率。

收敛分析。在图5中，我们全面分析了近期提出的各种方法的收敛情况。值得注意的是，在计算复杂度相近的情况下，我们的 InterFormer 优于其他最先进的方法，仅需四次初始点击便可达到 90% IoU。这种高效性主要得益于 InterFormer 所用 MAE 预训练视觉变换器的强大泛化特征。

对不同大小物体的评估。为展示 InterFormer 无论感兴趣物体大小如何都能保持一致的性能，我们对不同大小的物体进行了实验。在图 4中，我们展示了各模型针对面积比例大于不同阈值（横轴）的物体所得到的平均 NoC85 和 NoC90。我们将 InterFormer 与具有较低 SPC 的 FocalClick（如表 2所示）以及一种在大型物体情况下可能发生退化的调整大小策略进行了比较。结果表明，如图 4所示，在大型物体上，就 NoC85 和 NoC90 而言，InterFormer 的表现优于FocalClick。此外，InterFormer 在不同大小阈值下都能保持性能的一致性，进一步验证了其鲁棒性和多功能性。

此外，我们的研究结果凸显了在实际场景中使用 PIE 作为评估交互式分割模型指标的重要性。目前没有一种通用的缩放策略能够适用于所有尺寸的物体。因此，使用能够考量模型在不同尺寸物体上的表现的指标，对于评估其有效性至关重要。InterFormer 能够在不同尺寸的物体上保持性能，这凸显了它在实际应用中的价值。

4.3. 消融研究

我们进行了消融研究，以评估不同模型组件对 InterFormer 性能的影响。具体而言，我们将提出的 I-MSA 模块替换为一个类似普通 FPN 的模块，该模块将点击嵌入直接添加到大型编码器提取的特征中。我们还评估了采用 RITM 训练策略训练的模 [27] 型，以及采用更简单训练策略训练的模型的性能，并研究使用 ViT-Large 等更大型骨干网络的影响。为加快分析进程，我们将每个模型的训练时长缩短为 $40k$次迭代，而非标准的 $320k$ 次迭代。

表3 展示了我们全面的消融研究结果。研究发现，训练策略的选择对 Inter-

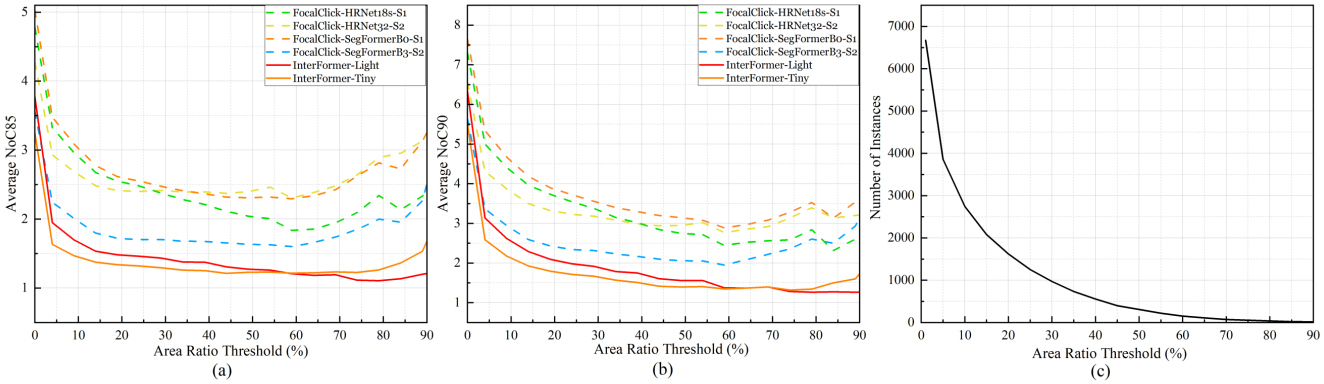


图 4. InterFormer 与 FocalClick 在不同尺寸物体上的比较，指标为面积比例超过不同阈值的物体的平均 NoC85（a）和 NoC90（b）。较高的阈值（例如 $> 80\%$）会筛除更多实例（c），从而导致 NoC 指标的估计产生噪声。这些结果表明，InterFormer 的性能优于 FocalClick，尤其是在较大的物体上，展现出更强的鲁棒性和适用性。

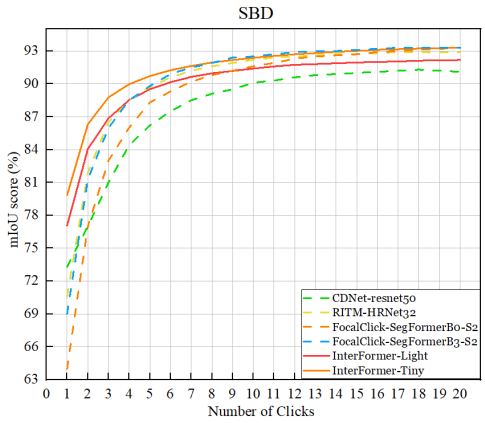


图 5. 交互式分割模型的收敛性分析。值得注意的是，我们的 InterFormer 展现出卓越的收敛性，超越了近期提出的、计算开销相近的其他方法，仅需四次点击便达到 90% 的 IoU。

Former 的性能影响甚微，而采用 ViT-Large 这样更大的骨干网络则能显著提升模型性能。值得注意的是，研究凸显了交互模块对于提升我们处理流程中交互式分割模型性能的关键作用。具体而言，我们的研究发现，使用类似 FPN 的标准交互模块会导致性能下降。然而，我们提出的 I-MSA 模块显著优于标准模块，表明它能有效提升交互式分割模型的性能。

4.4. 定性结果

我们对经过训练的 InterFormer- Tiny 进行了定性评估，训练时长为 $320k$ 次迭代。评估结果如图所示 6，其中 InterFormer 对三个

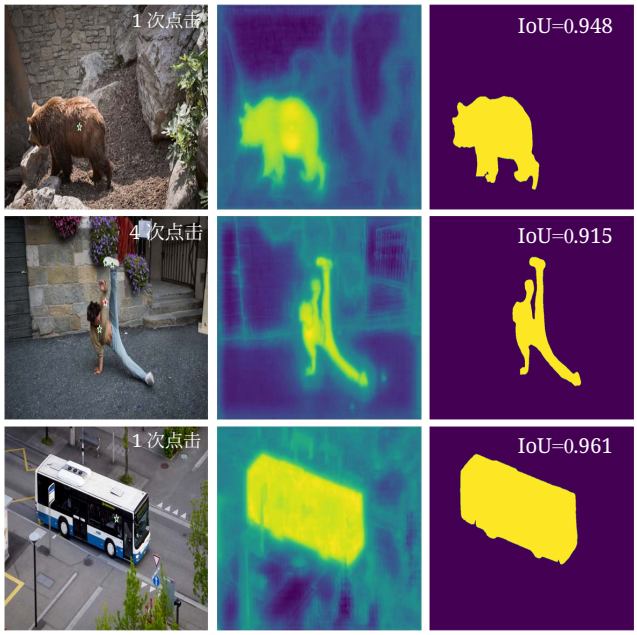


图 6. DAVIS 上的定性结果。第一列展示带有点击标记的原始图像，点击以五角星表示（绿色为正向点击，红色为负向点击）。第二列以热力图形式展示模型预测的 logits。第三列展示预测的分割结果，其中标示了与真实标注计算得到的 IoU。

仅需一次点击的情况。

5. Conclusion

本文致力于提升交互式图像分割的交互效率和速度。以往的方法通常会在每个交互步骤中重复计算图像特征。本文提出了InterFormer，这是一种新的处理流程，将图像处理与交互-

交互式分割。InterFormer 在高性能设备上使用大型视觉变换器（ViT）并行预处理图像，随后在低功耗设备上采用轻量级交互式多头自注意力（I-MSA）模块，实现实时分割。大量实验表明，InterFormer 在仅使用 CPU 的设备上也能实现高质量的交互式分割，在效率和质量方面均优于以往模型。

致谢。 本研究得到了国家重点研发计划（No.2022ZD0118202）、国家杰出青年科学基金（No.62025603）、国家自然科学基金（No. U21B2037, No. U22B2051, No. 62176222, No. 62176223, No. 62176226, No. 62072386, No. 62072387, No. 62072389, No. 62002305 and No. 62272401）以及福建省自然科学基金（No.2021J01002, No.2022J06001）的资助。

参考文献

[1] David Acuna、Huan Ling、Amlan Kar 和 Sanja Fidler。使用 polygon-rnn 高效交互式标注分割数据集++。载于 *2018* 年 *IEEE*计算机视觉与模式识别会议，*CVPR 2018*，美国犹他州盐湖城，*2018* 年 *6* 月 *18—22* 日，第 859–868 页。IEEE计算机学会，2018 年。1[2] Eirikur Agustsson、Jasper R. R. Uijlings 和 Vittorio Ferrari。通过联合考虑所有区域实现交互式全图像分割。载于 *IEEE*计算机视觉与模式识别会议，*CVPR 2019*，美国加利福尼亚州长滩，*2019* 年 *6* 月 *16—20* 日，第 11622–11631 页。计算机视觉基金会 / IEEE，2019 年。1[3] Saber Mirzaee Bafti、Chee Siang Ang、Md. Moinul Hossain、Gianluca Marcelli、Marc Alemany-Fornes 和 Anastasios D. Tsaousis。面向细胞生物学的众包半自动图像分割平台。2021 年。1[4] Junjie Bai 和 Xiaodong Wu。基于容错涂鸦的交互式图像分割。载于 *2014* 年 *IEEE*计算机视觉与模式识别会议，*CVPR 2014*，美国俄亥俄州哥伦布，*2014* 年 *6* 月 *23—28* 日，第 392–399 页。IEEE计算机学会，2014 年。1[5]Rodrigo Benenson、Stefan Popov 和 Vittorio Ferrari。与人工标注员协作进行大规模交互式目标分割。载于 *IEEE*计算机视觉与模式识别会议，*CVPR 2019*，美国加利福尼亚州长滩，*2019* 年 *6* 月 *16—20* 日，第 11700–11709 页。计算机视觉基金会 / IEEE，2019 年。1[6] Gedas Bertasius 和 Lorenzo Torresani。利用掩码传播对视频中的目标实例进行分类、分割和跟踪。载于 *2020* 年 *IEEE/CVF*计算机视觉与模式识别会议，*CVPR 2020*，美国华盛顿州西雅图，*2020* 年 *6* 月 *13—19* 日，第 9736–9745 页。IEEE，2020 年。1

[7] Yuri Y Boykov 和 M-P Jolly。用于 nd 图像中物体最优边界与区域分割的交互式图割。1:105–112，2001。2，6[8] Holger Caesar、Varun Bankiti、Alex H Lang、Sourabh Vora、Venice Erin Liong、Qiang Xu、Anush Krishnan、Yu Pan、Giancarlo Baldan 和 Oscar Beijbom。nuScenes：用于自动驾驶的多模态数据集。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第 11621–11631 页，2020。1[9] Xi C hen、Zhiyan Zhao、Feiwu Yu、Yilei Zhang 和 Manni Duan。用于交互式分割的条件扩散。第 7345–7354 页，2021。6[10] Xi Chen、Zhiyan Zhao、Yilei Zhang、Manni Duan、Donglian Qi 和 Hengshuang Zhao。FocalClick：迈向实用的交互式图像分割。第 1300–1309 页，2022。1，2，3，4，6，7[11] Dengxin Dai、Hayko Riemenschneider 和 Luc Van Gool。纹理示例的可合成性。载于 2014 IEEE计算机视觉与模式识别会议，CVPR 2014，美国俄亥俄州哥伦布，2014 年 6 月 23–28 日，第 3027–3034 页。IEEE计算机学会，2014。6[12] Alexey Dosovitskiy、Lucas Beyer、Alexander Kolesnikov、Dirk Weissenborn、Xiaohua Zhai、Thomas Unterthiner、Mostafa Dehghani、Matthias Minderer、Georg Heigold、Sylvain Gelly、Jakob Uszkoreit 和 Neil Houlsby。一张图像胜过 16×16 个词：用于大规模图像识别的 Transformer。载于第九届国际学习表征会议，ICLR 2021，2021 年。3 2021 2，3，4虚拟会议，奥地利，5 月 - 。OpenReview.net，。[13] Mark Everingham、Luc Van Gool、Christopher Williams、John Winn 和 Andrew Zisserman。PASCAL 视觉目标类别（VOC）挑战赛。《国际计算机视觉期刊》，2010。6[14] Haoqi Fan、Bo Xiong、Karttikeya Mangalam、Yanghao Li、Zhicheng Yan、Jitendra Malik 和 Christoph Feichtenhofer。多尺度视觉变换器。第 6824–6835 页，2021。3[15] Marco Forte、Brian Price、Scott Cohen、Ning Xu 和 Franc¸ois Piti´e。实现交互式分割 99% 的准确率。arXiv：计算机视觉与模式识别，2020。6[16] Leo Grady。用于图像分割的随机游走。《IEEE模式分析与机器智能汇刊》，2006。2，6[17] Varun Gulshan、Carsten Rother、Antonio Cri minisi、Andrew Blake 和 Andrew Zisserman。用于交互式图像分割的测地星凸性。载于第二十三届 IEEE计算机视觉与模式识别会议，CVPR 2010，美国加利福尼亚州旧金山，2010 年 6 月 13–18 日，第 3129–3136 页。IEEE计算机学会，2010。6[18] Agrim Gupta、Piotr Doll´ar 和 Ross B. Girshick。LVIS：大词汇量实例分割数据集。载于 IEEE计算机视觉与模式识别会议，CVPR 2019，美国加利福尼亚州长滩，2019 年 6 月 16–20 日，第 5356–5364 页。计算机视觉基金会/IEEE，2019。6[19] Yuying Hao、Yi Liu、Zewu Wu、Lin Han、Yizhou Chen、Guowei Chen、Lutao Chu、Shiyu Tang、Zhiliang Yu、Zeyu Chen 等。EdgeFlow：通过边缘引导流实现实用的交互式分

割。第 1551–1560 页，2021。6[20] Bharath Hariharan、Pablo Arbelaez、Lubomir D. Bourdev、Subhransu Maji 和 Jitendra Malik。来自逆检测器的语义轮廓。载于 Dimitris N. Metaxas、Long Quan、Alberto Sanfeliu 和 Luc Van Gool 编，IEEE国际计算机视觉会议，ICCV 2011，西班牙巴塞罗那，2011 年 11 月 6–13 日，第 991–998 页。IEEE计算机学会，2011。2，5，6，7[21] Kaiming He、Xinlei Chen、Saining Xie、Yanghao Li、Piotr Doll´ar 和 Ross Girshick。掩码自编码器是可扩展的视觉学习器。arXiv：计算机视觉与模式识别，2021。3，6[22] Kaiming He、Georgia Gkioxari、Piotr Doll´ar 和 Ross B. Girshick。Mask R-CNN。载于 IEEE国际计算机视觉会议，ICCV 2017，意大利威尼斯，2017 年 10 月 22–29 日，第 2980–2988 页。IEEE计算机学会，2017。5[23] Jie Hu、Liujuan Cao、Yao Lu、ShengChuan Zhang、Yan Wang、Ke Li、Feiyue Huang、Ling Shao 和 Rongrong Ji。ISTR：基于 Transformer 的端到端实例分割。arXiv 预印本 arXiv:2105.00637，2021。3[24] Jie Hu、Linyan Huang、Tianhe Ren、Shengchuan Zhang、Rongrong Ji 和 Liujuan Cao。只分割一次：迈向实时全景分割。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第 17819–17829 页，2023。3[25] Won-Dong Jang 和 Chang-Su Kim。通过反向传播细化方案实现交互式图像分割。载于 IEEE计算机视觉与模式识别会议，CVPR 2019，美国加利福尼亚州长滩，2019 年 6 月 16–20 日，第 5297–5306 页。计算机视觉基金会/IEEE，2019。2，3，6[26] Noel E. O’ Connor Kevin McGuinness。交互式分割算法的比较评估。《模式识别》，2010。2，5，6[27] Anton Konushin Konstantin Sofiiuk、Ilia A. Petrov。通过掩码引导复兴交互式分割的迭代训练。arXiv：计算机视觉与模式识别，2021。2，3，4，5，6，7[28] Yanghao Li、Hanzi Mao、Ross Girshick 和 Kaiming He。探索用于目标检测的普通视觉Transformer骨干网络。3[29] Zhuwen Li、Qifeng Chen 和 Vladlen Koltun。具有潜在多样性的交互式图像分割。载于 2018 IEEE计算机视觉与模式识别会议，CVPR 2018，美国犹他州盐湖城，2018 年 6 月 18–22 日，第 577–585 页。IEEE计算机学会，2018。6[30] Jun Hao Liew、Scott Cohen、Brian Price、Long Mai 和 Jiashi Feng。深度交互式细长目标选择。第 305–314 页，2021。6[31] Tsung-Yi Lin、Piotr Doll´ar、Ross B. Girshick、Kaiming He、Bharath Hariharan 和 Serge J. Belongie。用于目标检测的特征金字塔网络。载于 2017 IEEE计算机视觉与模式识别会议，CVPR 2017，美国夏威夷州檀香山，2017 年 7 月 21–26 日，第 936–944 页。IEEE计算机学会，2017。2，3

[32] Tsung-Yi Lin、Michael Maire、Serge Belongie、J ames Hays、Pietro Perona、Deva Ramanan、Piotr Doll´ ar 和 C. Lawrence Zitnick。Microsoft COCO：上下文中的常见物体。《计算机科学讲义》，2014。6[33] Zheng Lin、Zheng-Peng Duan、Zhao Zhang、Chun-Le Guo 和 Ming-Ming Cheng。Focuscut：在交互式分割中深入聚焦视图。第 2637–2646 页，2022。1、3、4、6[34] Zheng Lin、Zhao Zhang、Lin-Zhuo Chen、Ming-Ming Cheng 和 Shao-Ping Lu。利用首次点击注意力进行交互式图像分割。载于《2020 IEEE/CVF计算机视觉与模式识别会议》，CVPR 2020，美国华盛顿州西雅图，2020 年 6 月 13–19 日，第 13336–13345 页。IEEE，2020。2、3、6[35] Huan Ling、Jun Gao、Amlan Kar、Wenzheng Chen 和 Sanja Fidler。利用 curve-gcn 快速进行交互式物体标注。载于《IEEE计算机视觉与模式识别会议》，CVPR 2019，美国加利福尼亚州长滩，2019 年 6 月 16–20 日，第 5257–5266 页。计算机视觉基金会 / IEEE，2019。1[36] Geert Litjens、Thijs Kooi、Babak Ehteshami Bejnordi、Ar- naud Arindra Adiyoso Setio、Francesco Ciompi、Mohsen Ghafoorian、Jeroen Awm Van Der Laak、Bram Van Gin- neken 和 Clara I S´anchez。深度学习在医学图像分析中的综述。《医学图像分析》，42 88，2017 60 1 : – 。[37] Qin Liu、Zhenlin Xu、Gedas Bertasius 和 Marc Niethammer。Simpleclick：使用简单视觉 Transformer 进行交互式图像分割。arXiv 预印本 arXiv:2210 11006，2022 3、6 。[38] Qin Liu、Meng Zheng、Benjamin Planche、Srikrishna Karanam、Terrence Chen、Marc Niethammer 和 Ziyan Wu。Pseudoclick：利用点击模仿进行交互式图像分割。第 728–745 页，2022。1、3、4、6[39] Ze Liu、Yutong Lin、Yue Cao、Han Hu、Yixuan Wei、Zheng Zhang、Stephen Lin 和 Baining Guo。Swin Transformer：使用移位窗口的分层视觉 Transformer。载于《IEEE/CVF国际计算机视觉会议论文集》，第

10012 10022, 2021 2, 3, 4, 5 – 。[40] Soumajit Majumder 和 Angela Yao。面向交互式实例分割的内容感知多级引导。载于《IEEE/CVF计算机视觉与模式识别会议论文集》，第 11602–11611 页，2019。6[41] Kevis-Kokitsi Maninis、Sergi Caelles、Jordi Pont-Tuset 和 Luc Van Gool。Deep extreme cut：从极值点到物体分割。载于《2018 IEEE计算机视觉与模式识别会议》，CVPR 2018，美国犹他州盐湖城，2018 年 6 月 18–22 日，第 616–625 页。IEEE计算机学会，2018。1、2、3[42]Federico Perazzi、Jordi Pont-Tuset、Brian McWilliams、Luc Van Gool、Markus H. Gross 和 Alexander Sorkine- Hornung。视频物体分割的基准数据集与评估方法。载于《2016 IEEE计算机视觉与模式识别会议》，CVPR 2016，美国内华达州拉斯维加斯，2016 年 6 月 27–30 日，第 724–732 页。IEEE计算机学会，

2016 2, 5, 6 。Carsten Rother、[43] Vladimir Kolmogorov 和 Andrew Blake。“GrabCut”：通过迭代进行交互式前景提取。

图割。《ACM图形学汇刊》(TOG)，23(3):309–314，2004。1, 2, 5, 6[44] Konstantin Sofiiuk、Ilia A. Petrov、Olga Barinova 和 Anton Konushin。F-BRS：重新思考用于交互式分割的反向传播细化。载于《2020年IEEE/CVF计算机视觉与模式识别会议》，CVPR 2020，美国华盛顿州西雅图，2020年6月13日至19日，第8620–8629页。IEEE，2020。2, 3, 5, 6[45] Robin Strudel、Ricardo Garcia、Ivan Laptev 和 Cordelia Schmid。Segmenter：用于语义分割的Transformer。第7262–7272页，2021。3[46] Chen Sun、Abhinav Shrivastava、Saurabh Singh 和 Abhinav Gupta。重新审视深度学习时代数据的惊人效用。载于《IEEE国际计算机视觉会议》，ICCV 2017，2017年10月29日至11月1日，意大利威尼斯，第843–852页。IEEE计算机学会，2017。1[47] Hugo Touvron、Matthieu Cord、Matthijs Douze、Francisco Massa、Alexandre Sablayrolles 和 Herv´e J´egou。通过注意力训练数据高效的图像Transformer并进行蒸馏。载于 Marina Meila 和 Tong Zhang 编，《第38届国际机器学习会议论文集》，ICML 2021，2021年7月18日至24日，线上会议，《机器学习研究论文集》第139卷，第10347–10357页。PMLR，2021。3[48] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N. Gomez、Lukasz Kaiser 和 Illia Polosukhin。注意力就是一切。载于 Isabelle Guyon、Ulrike von Luxburg、Samy Bengio、Hanna M. Wallach、Rob Fergus、S. V. N. Vishwanathan 和 Roman Garnett 编，《神经信息处理系统进展30：2017年神经信息处理系统年度会议》，2017年12月4日至9日，美国加利福尼亚州长滩，第5998–6008页，2017。3[49] Wenhai Wang、Enze Xie、Xiang Li、Deng-Ping Fan、Kaitao Song、Ding Liang、Tong Lu、Ping Luo 和 Ling Shao。金字塔视觉Transformer：一种无需卷积、适用于密集预测的通用骨干网络。第568–578页，2021。2, 3, 4[50] Jiajun Wu、Yibiao Zhao、Jun-Yan Zhu、Siwei Luo 和 Zhuowen Tu。Milcut：一种用于交互式图像分割的扫描线多实例学习范式。载于《2014年IEEE计算机视觉与模式识别会议》，CVPR 2014，美国俄亥俄州哥伦布，2014年6月23日至28日，第256–263页。IEEE计算机学会，2014。1[51] Yu-Huan Wu、Yun Liu、Xin Zhan 和 Ming-Ming Cheng。P2T：用于场景理解的金字塔池化Transformer。《IEEE模式分析与机器智能汇刊》，2022。2, 3, 5[52] Tete Xiao、Yingcheng Liu、Bolei Zhou、Yuning Jiang 和 Jian Sun。用于场景理解的统一感知解析。第418–434页，2018。4[53] Enze Xie、Wenhai Wang、Zhiding Yu、Anima Anandkumar、Jose M Alvarez 和 Ping Luo。SegFormer：基于Transformer的简单高效语义分割设计。《神经信息处理系统进展》，34:12077–12090，2021。3[54] Ning Xu、Brian L. Price、Scott Cohen 和 Thomas S. Huang。深度图像抠图。载于《2017年IEEE计算机视觉与模式识别会议》。

视觉与模式识别，计算机视觉与模式识别会议 *2017*，檀香山，美国夏威夷州，*2017*年*7*月*21*日至*26*日，第311–320页。IEEE计算机学会，2017年。6[55] Ning Xu、Brian L. Price、Scott Cohen、Jimei Yang 和 Thomas S. Huang。深度交互式对象选择。载于*2016*年 *IEEE* 计算机视觉与模式识别会议，*CVPR 2016*，美国内华达州拉斯维加斯，*2016*年*6*月*27*日至*30*日，第373–381页。IEEE计算机学会，2016年。2，6[56] Li Yuan、Yunpeng Chen、Tao Wang、Weihao Yu、Yujun Shi、Zi-Hang Jiang、Francis EH Tay、Jiashi Feng 和 Shuicheng Yan。Tokens-to-token ViT：从零开始在 ImageNet 上训练视觉 Transformer。第558–567页，2021年。3[57] Shiyin Zhang、Jun Hao Liew、Yunchao Wei、Shikui Wei 和 Yao Zhao。利用内外部引导进行交互式对象分割。载于*2020*年 *IEEE/CVF* 计算机视觉与模式识别会议，*CVPR 2020*，美国华盛顿州西雅图，*2020*年*6*月*13*日至*19*日，第12231–12241页。IEEE，2020年。1[58] Sixiao Zheng、Jiachen Lu、Hengshuang Zhao、Xiatian Zhu、Zekun Luo、Yabiao Wang、Yanwei Fu、Jianfeng Feng、Tao Xiang、Philip HS Torr 等。以序列到序列的视角重新思考语义分割：采用 Transformer。第6881–6890页，2021年。3