

PartSLIP: Low-Shot Part Segmentation for 3D Point Clouds via Pretrained Image-Language Models

摘要

通用的3D部件分割在视觉和机器人领域具有重要意义，但也面临挑战。通过传统的监督方法训练深度模型需要大规模的3D数据集，并需要详细的部件注释，这在收集过程中成本高昂。本文探讨了一种替代方法，通过利用预训练的图像语言模型GLIP，在开放词汇2D检测任务上取得卓越性能，来实现对3D点云的小样本部件分割。我们通过GLIP在点云渲染上进行基于部件的检测以及一种新的2D到3D标签提升算法，将从2D中获得的丰富知识迁移到3D。同时，我们还利用多视图3D先验和小样本提示调整来显著提升性能。在PartNet和PartNet-Mobility数据集上的广泛评估表明，我们的方法实现了出色的零样本3D部件分割。我们的小样本版本不仅在很大程度上优于现有的小样本方法，而且与完全监督的对应方法相比，取得了高度竞争力的结果。此外，我们证明我们的方法可以直接应用于iPhone扫描的点云，而且没有明显的领域间隙。

1 引言

虽然传统的监督训练使各种基于数据的3D部件分割方法能够取得显著的结果，但它们在处理未见过的类别形状时常常遇到困难。与图像数据集相比，带有3D部件标注的数据集在规模上仍然小得多。例如，最近的PartNet数据集仅包含24个对象类别，远远少于现实世界中对象类别的数量。提供足够的训练数据以涵盖所有对象类别是一项具有挑战性的任务，因此，传统的监督训练方法往往缺乏泛化性。

最近，图像-文本大模型出色的性能吸引了越来越多研究者的关注。通过在大规模图像-文本对上进行预训练，图像-语言大模型学习了广泛的视觉概念和知识，它们可以被自然语言引用。由于它们令人印象深刻的零样本能力，它们已经被广泛用于实现各种2D/3D视觉和语言任务。在这篇论文中，作者致力于利用预训练的图像-语言大模型GLIP来解决零样本和小样本的3D部件分割问题。

2 相关工作

2.1 3D部件分割

3D部件分割涉及两个主要任务：语义分割和实例分割。大多数3D网络都能够通过为每个几何基元(例如点或体素)预测语义标签来进行语义分割。现有的基于学习的方法通过将各种分组或区域提议策略纳入管道中来解决实例分割问题。与使用每个点部分标签的标准训练

不同，一些工作利用弱监督，例如包围盒，语言参考游戏或IKEA手动操作。还有一些工作不仅关注单个对象，还考虑了场景尺度输入的部件分割。此外，与语义和实例分割的两个经典任务不同，另一条平行的研究线将3D形状分解为一组分层细粒度部件，但不考虑语义标签，这与我们的目标不同。最近，一些工作还提出学习连续的隐式语义场。

2.2 利用图像-文本大模型进行3D学习

最近，预训练的图像语言模型通过在大规模图像文本对上进行预训练取得了显著的进展。由于它们学到了丰富的视觉概念和令人印象深刻的零样本能力，它们已经被应用到各种3D视觉任务中，如3D头像生成和操作，通用3D形状生成，低样本3D形状分类，神经辐射场，3D视觉引导以及3D表示学习等。本文首次利用预训练图像语言模型来协助3D部件分割任务的研究。

2.3 GLIP简介

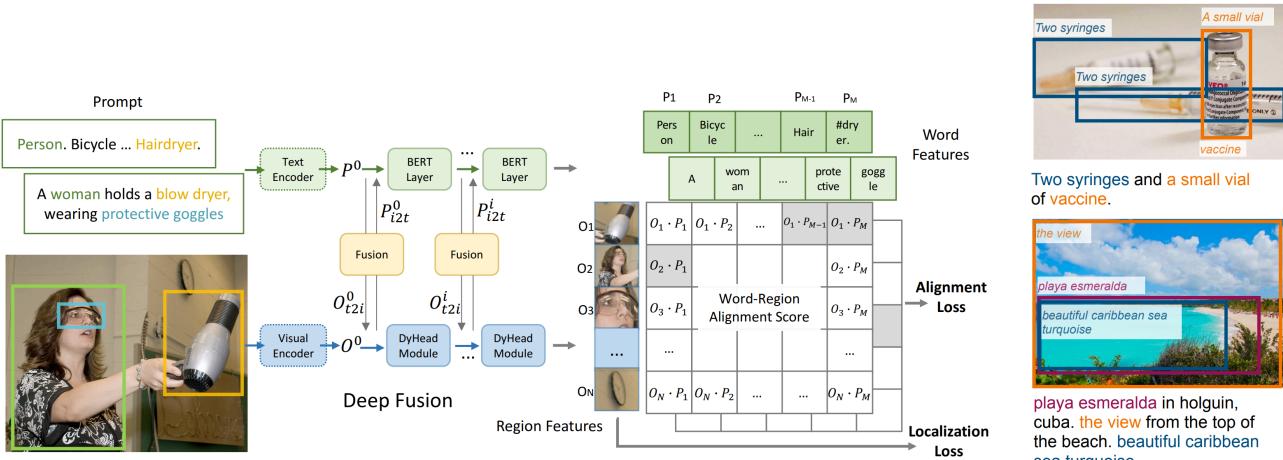


图 1. GLIP结构图

图像-语言大模型GLIP [3]专注于2D视觉定位和探测任务。它以自由形式的文本描述和2D图像作为输入，并通过为输入图像输出多个2D包围盒来定位文本中的所有短语。通过在大规模图像-文本对(27M数据)上进行预训练，GLIP模型学习了广泛的视觉概念，并实现了开放词汇的2D探测。

3 本文方法

3.1 输入输出

输入：带有RGB信息的3D点云和包含感兴趣部件名称的文本提示
输出：3D点云的语义分割和实例分割

3.2 方法概述

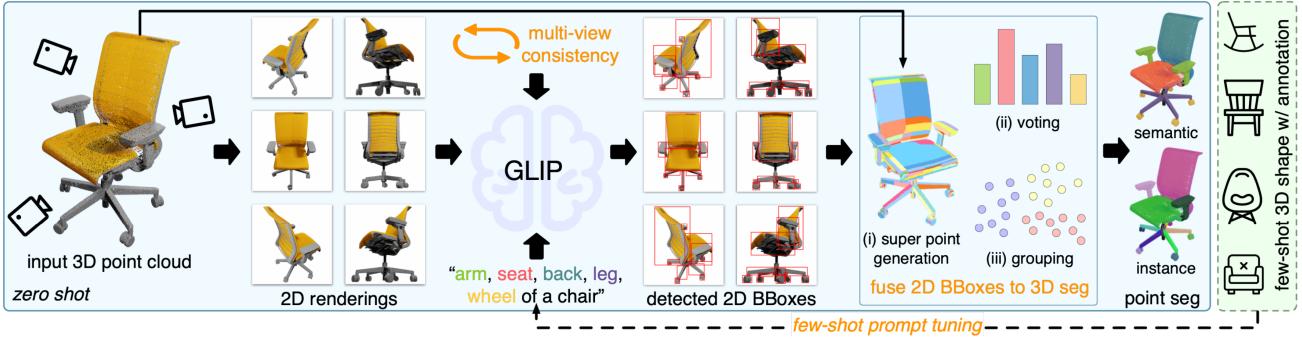


图 2. PartSLIP结构图

这篇论文 [4]的方法以3D点云和文本提示作为输入，并以零样本或小样本的方式生成3D语义和实例分割。该方法利用了图像-语言大模型GLIP的2D视觉定位能力，具体步骤如下：(1)为了将3D输入与2D GLIP模型连接起来，作者为输入点云渲染多视角的2D图像，然后将其与包含感兴趣部件名称的文本提示一起输入到GLIP模型中；(2)GLIP模型检测每个2D视图中感兴趣的部件，以2D包围盒的形式输出检测结果；(3)利用3D投票和分组模块，融合多视角的2D包围盒，为点云生成3D语义和实例分割。

3.3 投票和分组模块

由于将2D包围盒转换回3D语义和实例分割并不是一件简单的事情，论文提出了一个新的3D投票和分组模块，以融合多视角的2D包围盒，并为输入点云生成3D实例分割。该模块包含三个步骤：(1)引用 [2]的方法，对输入点云进行聚类生成超点；(2)依据超点出现在每个视图包围盒中的情况，为每个超点分配一个部件类别标签；(3)为超点分组以生成部件实例，分组标准包括：(a)两个超点是否具有相同的语义标签、(b)是否在位置上相邻、(c)是否同时在某个包围盒内或包围盒外。

3.4 小样本微调

有时，预训练的GLIP模型可能无法完全通过文本提示理解我们对部件的定义。论文中提到的一个有效的解决方案是使用带有部件分割的3D形状小样本进行提示调整：在固定预训练的GLIP模型的参数的同时，学习每个部件名称的语言嵌入的偏移特征向量。此外，作者还提出了一个多视角视觉特征聚合模块，以融合多个2D视图的信息，这样GLIP模型可以更好地理解输入3D形状的全局信息。

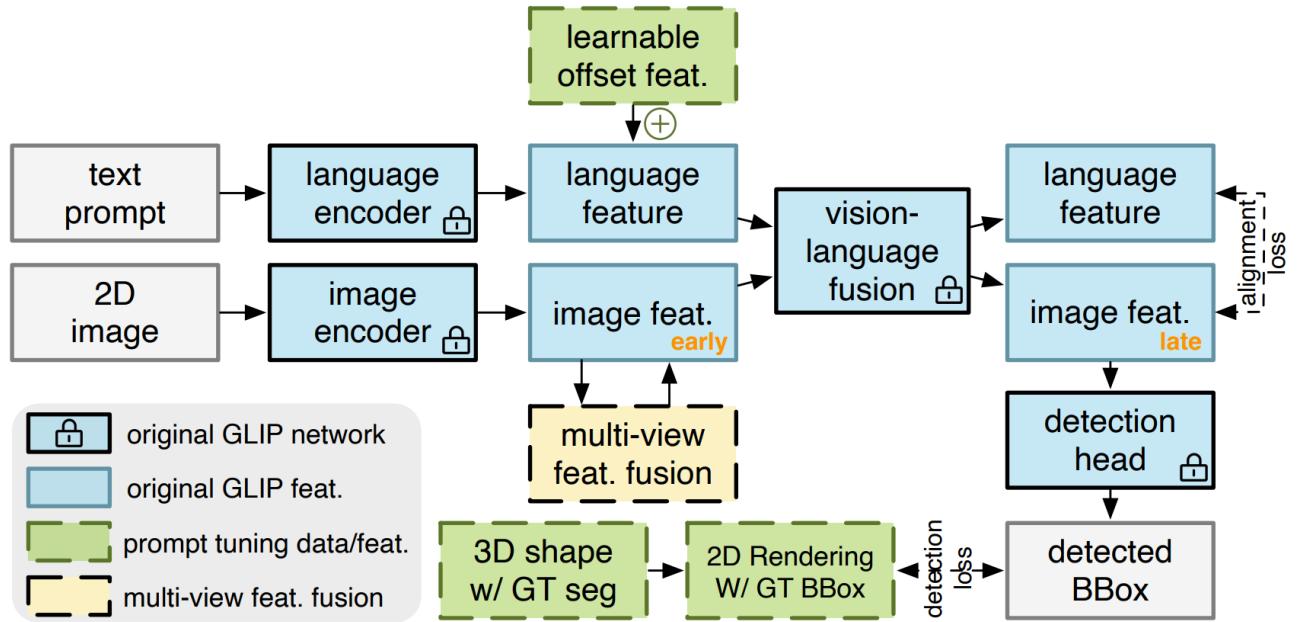


图 3. 小样本提示微调和多视角特征融合

如图 3 所示，带锁的蓝色框表示 GLIP 原始的网络结构模块，在微调的时候冻结其参数。为了引入可以微调的参数，learnable offset feat. 模块为语言特征预测一个残差偏移量，multi-view feat. fusion 模块为图像特征预测多视角融合的权重。也就是说，输入 vision-language fusion 模块的是加上偏移量的语言特征和加权求和的图像特征。这样，网络就多了两个可学习的模块，可以使用小样本进行微调。

3.5 论文中的实验结果

3.5.1 语义分割

#3D data	method	Overlapping Categories										Non-Overlapping Categories										Overall (17)	Overall (45)
		Bottle	Chair	Display	Door	Knife	Lamp	Storage Furniture	Table	Overall	Camera	Cart	Dis-	Penser	Kettle	Oven	Suit-	case	Toaster	Overall			
few-shot w/ extra data (45x8+28k)	PointNet++ [43]	48.8	84.7	78.4	45.7	35.4	68.0	46.9	63.7	55.6	6.5	6.4	12.1	20.9	15.8	34.3	40.6	14.7	25.4	36.8	45.1	50.2	
	PointNext [44]	68.4	91.8	89.4	43.8	58.7	64.9	68.5	52.1	58.5	33.2	36.3	26.0	45.1	57.0	37.8	13.5	8.3	45.1	50.2			
	SoftGroup [58]	41.4	88.3	62.1	53.1	31.3	82.2	60.2	54.8	50.2	23.6	23.9	18.9	57.4	45.5	13.6	18.3	26.4	30.7	38.1			
few-shot (45x8)	PointNet++ [43]	27.0	42.2	30.2	20.5	22.2	10.5	8.4	7.3	18.1	9.7	11.6	7.0	28.6	31.7	19.4	3.3	0.0	21.8	20.4	41.1	38.0	
	PointNext [44]	67.6	65.1	53.7	46.3	59.7	55.4	20.6	22.1	39.2	26.0	47.7	22.6	60.5	66.0	36.8	14.5	0.0	41.5	40.6			
	SoftGroup [58]	20.8	80.5	39.7	16.3	38.3	38.3	18.9	24.9	32.8	28.6	40.8	42.9	60.7	54.8	35.6	29.8	14.8	41.1	38.0			
	ACD [12]	22.4	39.0	29.2	18.9	39.6	13.7	7.6	13.5	19.2	10.1	31.5	19.4	40.2	51.8	8.9	13.2	0.0	25.6	23.2			
	Prototype [80]	60.1	70.8	67.3	33.4	50.4	38.2	30.2	25.7	41.1	32.0	36.8	53.4	62.7	63.3	36.5	35.5	10.1	46.3	44.3			
zero-shot	Ours	83.4	85.3	84.8	40.8	65.2	66.0	53.6	42.4	56.3	58.3	88.1	73.7	77.0	69.6	73.5	70.4	60.0	61.3	59.4	24.4	27.2	
	Ours	76.3	60.7	43.8	2.7	46.8	37.1	29.4	47.7	31.8	21.4	87.7	16.5	20.8	4.7	33.0	40.2	13.8	24.4	27.2			

图 4. 语义分割结果

图 4 显示了语义分割的结果。论文的方法在一些常见的物体类别(如瓶子、椅子和桌子)上取得了令人瞩目的零样本性能，但在某些类别(例如水壶)上表现较差。这主要是因为预训练的 GLIP 模型可能无法理解文本提示的含义(例如水壶的喷嘴)。经过使用 8 个样本的 3D 数据进行提示调整后，论文的方法实现了 59.4% 的 mIoU，并在小样本设置以及 45 × 8 + 28k 设置中胜过所有基线方法。对于 45 × 8 + 28k 设置，基线方法使用额外的 28k 个形状进行训练，涵盖 17 个

类别。对于这些重叠的类别，这是一个全监督的设置，但论文方法的8样本版本可以实现非常有竞争力的总体mIoU(56.3% vs. 58.5%)。但是，28k的训练数据对于基线方法泛化到非重叠类别的帮助有限。论文的方法在非重叠类别上远远优于所有基线方法。



图 5. 语义分割结果可视化

由于GLIP模型具有强大的泛化能力，论文的方法可以直接在现实世界中部署。如图 6所示，使用配备LiDAR传感器的iPhone 12 Pro Max捕捉视频，并将融合的点云作为输入，可以观察到与合成数据集实验中类似的结果。



图 6. 扫描数据的语义分割结果

3.5.2 实例分割

#3D data	method	Overlapping Categories												Non-Overlapping Categories											
		Bottle	Chair	Display	Door	Knife	Lamp	Furniture	Table	Overall (17)	Storage	Dis-	Cart	Penser	Kettle	Cook-	Suit-	Pot	Oven	case	Toaster	Overall (28)	Overall (45)		
45x8+28k	PointGroup [23]	38.2	87.6	65.1	23.4	19.3	62.7	49.1	46.4	41.7	8.6	29.2	24.0	61.3	59.4	13.8	15.6	7.0	24.6	31.0					
	SoftGroup [58]	43.9	89.1	68.7	21.2	27.2	63.3	49.1	46.2	42.4	0.7	28.4	26.4	63.8	59.3	16.4	13.5	7.5	25.6	31.9					
few-shot (45x8)	PointGroup [23]	8.0	77.2	16.7	3.7	15.6	9.8	0.0	0.0	14.6	4.7	28.5	30.7	52.1	57.0	0.0	0.0	0.0	16.8	16.0					
	SoftGroup [58]	22.4	87.7	27.5	5.6	10.3	19.4	11.6	14.2	21.3	11.2	29.8	37.8	63.4	65.7	10.4	8.0	10.7	28.4	25.7					
	Ours	79.4	84.4	82.9	17.9	43.9	68.3	32.8	32.3	42.5	36.8	83.3	63.5	75.4	70.5	64.5	44.9	38.4	46.2	44.8					
zero-shot	Ours	75.5	54.5	32.9	1.3	22.1	35.8	10.9	36.6	20.9	8.4	79.3	9.3	18.3	1.1	25.9	34.2	4.5	16.2	18.0					

图 7. 语义分割结果

图 7 显示了实例分割的结果。与语义分割的结果类似，论文的方法在零样本设置中实现了 18.0% 的 mAP50，在 8 样本设置中实现了 44.8% 的 mAP50，优于 45×8 和 $45 \times 8 + 28k$ 设置中的所有基线方法。

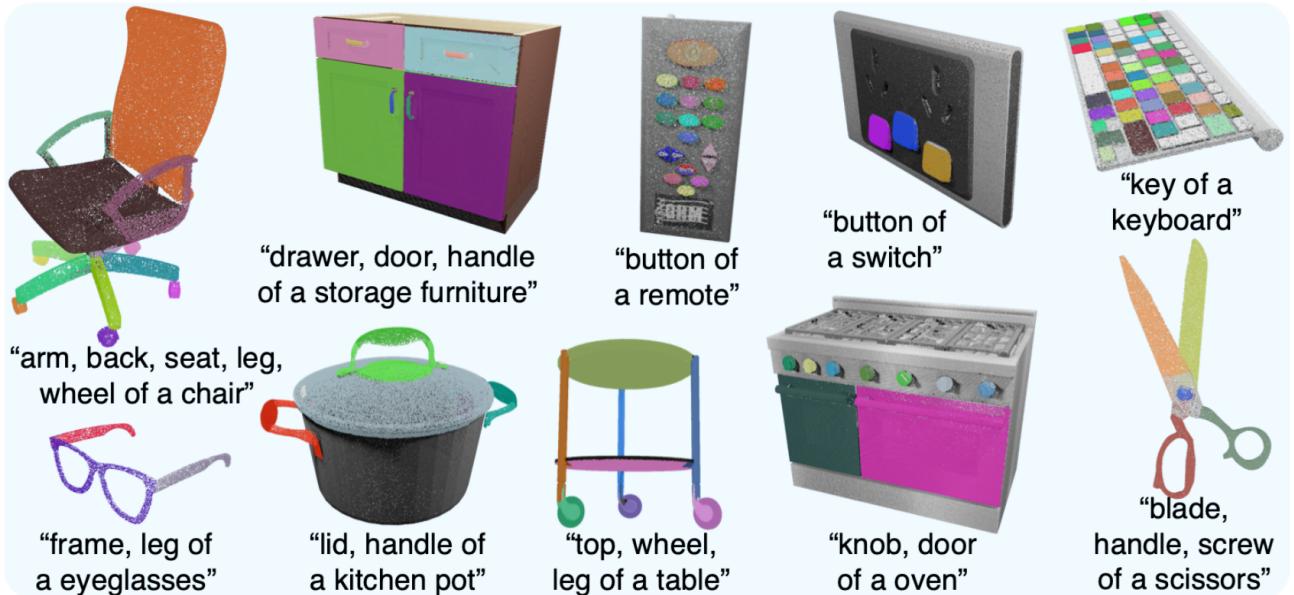


图 8. 语义分割结果可视化

4 复现细节

4.1 实验环境搭建

作者已经提供了项目源代码，链接如下：<https://github.com/ricklentz/PartSLIP>，由于项目依赖的部分代码并不是跨平台的，因此建议使用 Ubuntu 18.04 及以上版本。如果使用的是 Win10 或者是 Win11 的系统，那么大概率会遇到 cmake 编译的代码无法链接的问题。

4.1.1 创建 Conda 环境并安装依赖项

项目中包含了 environment.yml 文件用于定义和管理 Python 环境，通过它可以直接创建和配置项目所需要的 Conda 环境。由于在 environment.yml 文件中配置的 cudatoolkit 的版本是 11.2，

在安装CUDA时也应该安装11.2版本，安装其他CUDA版本可能导致与PyTorch版本不匹配的问题。通过以下指令安装Conda环境并激活：

```
conda env create -f environment.yml  
conda activate partslip
```

4.1.2 安装PyTorch3D

论文的方法需要渲染点云得到3D形状的多视角图像，在它的项目代码中是通过PyTorch3D实现的。注意，PyTorch3D版本与PyTorch版本必须对应。通过以下指令安装PyTorch3D：

```
pip install "git+https://github.com/facebookresearch/pytorch3d.git"
```

4.1.3 安装cut-pursuit

论文的方法利用cut-pursuit来计算超点。通过以下指令安装cut-pursuit：

```
git submodule update --init  
cd GLIP  
python setup.py build develop --user
```

4.1.4 下载预训练的网络参数

首先，需要下载GLIP的预训练参数。此外，对于小样本设置，作者也提供了对各个对象类别进行微调得到的训练参数，可以从指定链接下载。

4.1.5 示例程序

作者提供了5个示例点云文件和1个示例程序。在下载5+1个网络参数文件后，可以用下面的指令运行零样本和小样本推断：

```
python3 demo.py
```

4.1.6 示例程序运行结果

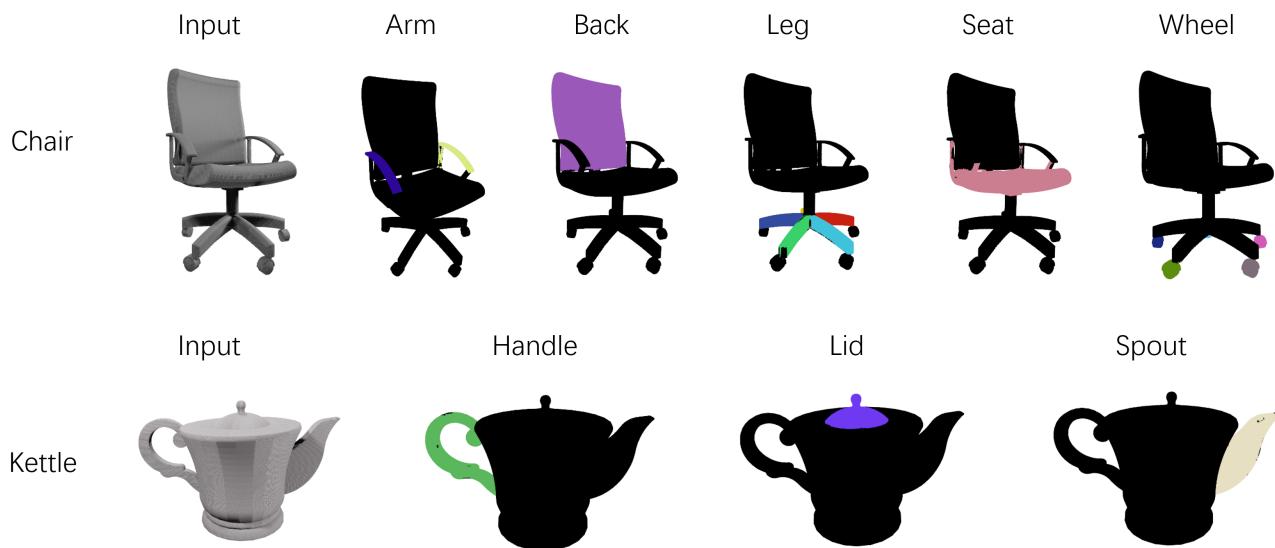


图 9. few-shot实例分割

4.2 在不同运动状态的物体上测试

4.2.1 生成不同运动状态的点云

点云旋转变换：(a)首先把旋转轴枢轴点移到原点，点云跟着移动;(b)依据旋转轴和旋转角度构造表示旋转的四元数；(c)把四元数转为旋转矩阵，与指定部件点云坐标相乘得到旋转后的点云;(d)把旋转轴枢轴点移回原位，点云跟着移动。

```
def rotate_part(pts, pointIndicator, joint_dir, joint_pos, angle):
    """
    rotate the moving part along (joint_dir, joint_pos) with angle.
    """
    part_pts = pts[pointIndicator == 1]
    part_pts -= joint_pos
    motion_quat = np.hstack([
        [np.cos(angle / 360 * 2 * 3.14 / 2), np.sin(angle / 360 * 2 * 3.14 / 2) * joint_dir]
    ])
    part_pts = \
        rotate_module(torch.from_numpy(part_pts).view(
            1, -1, 3), torch.from_numpy(motion_quat).view(1, 1, 4)).numpy()[0]
    part_pts += joint_pos
    pts[pointIndicator == 1] = part_pts
    return pts
```

图 10. 绕旋转轴旋转固定角度

点云平移变换：(a)旋转轴方向乘以长度得到平移向量;(b)指定部件点云坐标加上平移向量得到平移后的点云。

```

def translate_part(pts, pointIndicator, joint_dir, amount):
    """
    translate the moving part along the translation axis with amount.
    """
    part_pts = pts[pointIndicator == 1]
    part_pts += (joint_dir * amount)
    pts[pointIndicator == 1] = part_pts
    return pts

```

图 11. 沿平移轴平移固定长度

4.2.2 实例分割结果

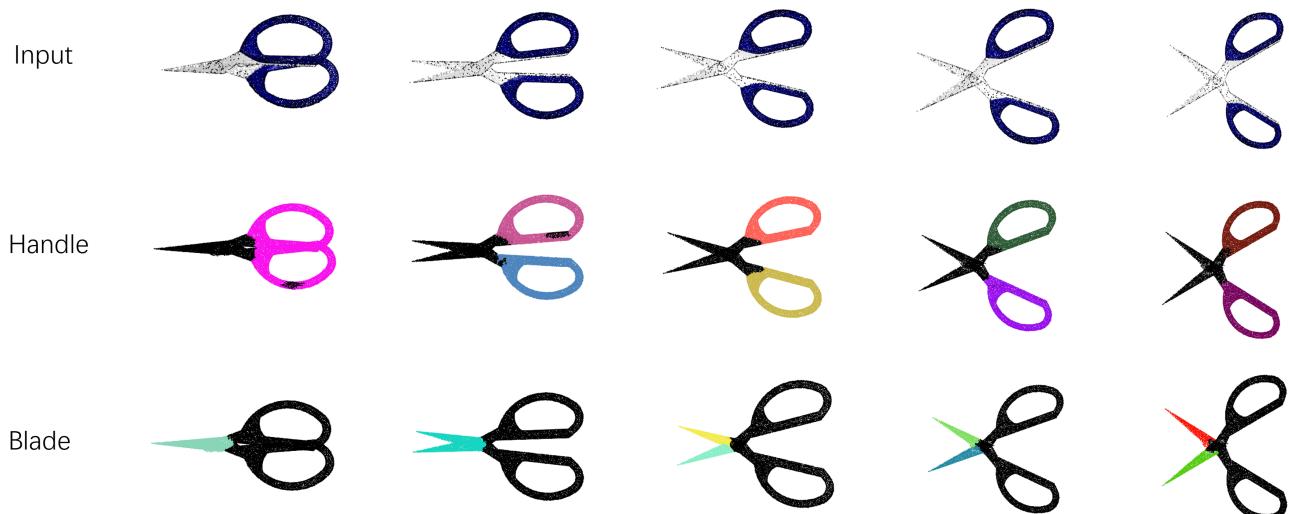


图 12. 不同运动状态剪刀的实例分割

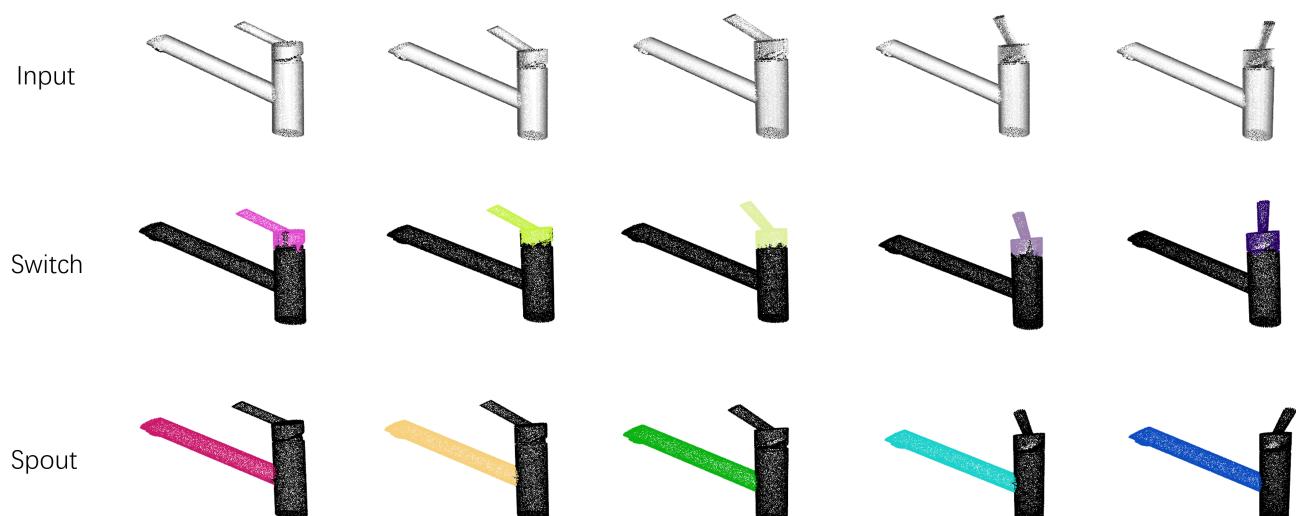


图 13. 不同运动状态水龙头的实例分割

5 改进思路

5.1 改进方法概述

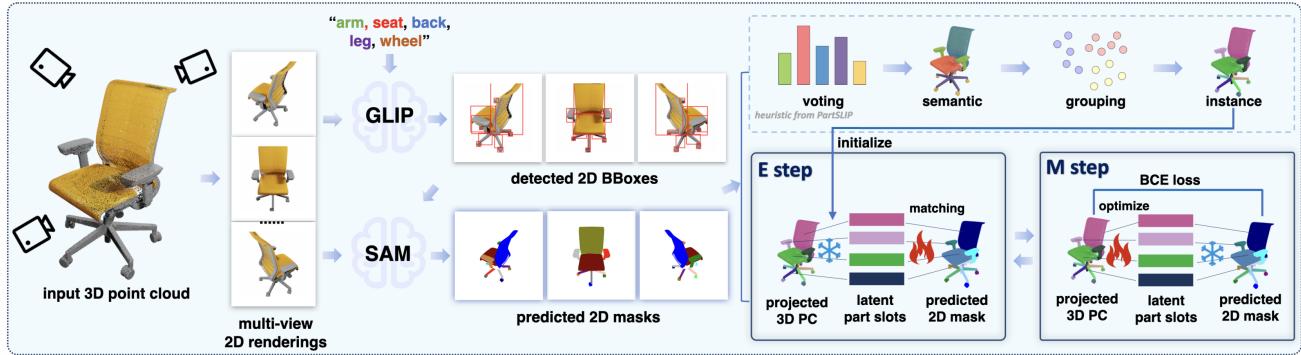


图 14. PartSLIP++结构图

与PartSLIP相同，PartSLIP++ [5]首先以密集的3D点云作为输入。它首先为点云渲染多视图图像。这些图像连同文本提示一起输入GLIP模型，预测2D包围盒。不同的是，随后PartSLIP++利用SAM模型，根据预测的2D包围盒作为提示，为每个视图生成2D实例分割掩码。这些多视图2D实例掩码使用新的修改后的EM算法转换为3D部分分割掩码。在E步中，通过匈牙利算法找到投影的3D分割和2D预测实例掩码之间的最佳匹配。在M步中，找到的匹配用于通过梯度下降优化改进3D分割。EM算法需要一个初始条件，PartSLIP++使用PartSLIP提出的3D投票和分组模块用于初始化3D实例分割。

5.2 改进部分总结

(1) 将预训练的2D分割模型SAM [1]整合到PartSLIP方法中，产生比先前工作中使用的包围盒更准确和精确的2D像素级部件注释。

(2) 将多视图2D部件分割提升为3D部件分割的问题看作最大似然估计问题，引入一种新的修改过的期望最大化(EM)算法，以有效优化这个问题。

5.3 改进结果

5.3.1 语义分割

#3D data	method	Overlapping Categories										Non-Overlapping Categories										Overall (45)
		Bottle	Chair	Display	Door	Knife	Lamp	Furniture	Table	Overall (17)	Dis-	Cart	Penser	Kettle	Kitchen-	Pot	Oven	Suit-	case	Toaster	Overall (28)	
											Camera											
Few-shot w/ extra data (45x8+28k)	PointNet++ [31]	48.8	84.7	78.4	45.7	35.4	68.0	46.9	63.7	55.6	6.5	6.4	12.1	20.9	15.8	34.3	40.6	14.7	25.4	36.8		
	PointNext [32]	68.4	91.8	89.4	43.8	58.7	64.9	68.5	52.1	58.5	33.2	36.3	26.0	45.1	57.0	37.8	13.5	8.3	45.1	50.2		
	SoftGroup [38]	41.4	88.3	62.1	53.1	31.3	82.2	60.2	54.8	50.2	23.6	23.9	18.9	57.4	45.5	13.6	18.3	26.4	30.7	38.1		
	PointNet++ [31]	27.0	42.2	30.2	20.5	22.2	10.5	8.4	7.3	18.1	9.7	11.6	7.0	28.6	31.7	19.4	3.3	0.0	21.8	20.4		
	PointNext [32]	67.6	65.1	53.7	46.3	59.7	55.4	20.6	22.1	39.2	26.0	47.7	22.6	60.5	66.0	36.8	14.5	0.0	41.5	40.6		
	SoftGroup [38]	20.8	80.5	39.7	16.3	38.3	38.3	18.9	24.9	32.8	28.6	40.8	42.9	60.7	54.8	35.6	29.8	14.8	41.1	38.0		
	ACD [8]	22.4	39.0	29.2	18.9	39.6	13.7	7.6	13.5	19.2	10.1	31.5	19.4	40.2	51.8	8.9	13.2	0.0	25.6	23.2		
Few-shot (45x8)	Prototype [50]	60.1	70.8	67.3	33.4	50.4	38.2	30.2	25.7	41.1	32.0	36.8	53.4	62.7	63.3	36.5	35.5	10.1	46.3	44.3		
	PartSLIP [21]	83.4	85.3	84.8	40.8	65.2	66.0	53.6	42.4	56.3	58.3	88.1	73.7	77.0	69.6	73.5	70.4	60.0	61.3	59.4		
	PartSLIP* [21]	81.2	82.7	81.8	43.1	62.5	66.3	52.3	44.3	56.6	61.8	79.0	71.0	73.3	66.5	69.1	64.5	50.1	58.7	57.9		
	Ours	85.8	85.3	85.1	45.1	64.3	67.9	57.2	45.3	57.0	63.2	84.8	72	85.6	76.8	70.3	70.0	50.7	63.3	60.8		

图 15. 语义分割结果

图 15 中展示了语义分割的结果。当在 PartNet-E 的 45×8 形状的小样本数据集上训练时，PartSLIP++ 相比先前的基线方法表现最好。特别是，在 PartNet-E 的 45 个类别中，它的性能比已发布的 PartSLIP 检查点高出 2.9 % 的 mIoU(60.8 vs. 57.9)。这些结果表明 PartSLIP++ 在小样本 3D 语义分割中的有效性。

5.3.2 实例分割

#3D data	method	Overlapping Categories										Non-Overlapping Categories										Overall (45)
		Bottle	Chair	Display	Door	Knife	Lamp	Furniture	Table	Overall (17)	Dis-	Cart	Penser	Kettle	Kitchen-	Pot	Oven	Suit-	case	Toaster		
											Camera											
45x8+28k	PointGroup [14]	38.2	87.6	65.1	23.4	19.3	62.7	49.1	46.4	41.7	8.6	29.2	24.0	61.3	59.4	13.8	15.6	7.0	24.6	31.0		
	SoftGroup [38]	43.9	89.1	68.7	21.2	27.2	63.3	49.1	46.2	42.4	0.7	28.4	26.4	63.8	59.3	16.4	13.5	7.5	25.6	31.9		
	PointGroup [14]	8.0	77.2	16.7	3.7	15.6	9.8	0.0	0.0	14.6	4.7	28.5	30.7	52.1	57.0	0.0	0.0	0.0	16.8	16.0		
	SoftGroup [38]	22.4	87.7	27.5	5.6	10.3	19.4	11.6	14.2	21.3	11.2	29.8	37.8	63.4	65.7	10.4	8.0	10.7	28.4	25.7		
	PartSLIP [21]	79.4	84.3	82.9	17.9	43.9	68.3	32.8	32.3	42.5	36.8	83.3	63.5	75.4	70.5	64.5	44.9	38.4	46.2	44.8		
	PartSLIP* [21]	74.4	79.3	64.2	14	43.3	69.5	29.2	32.1	41.1	29.6	71	59.7	72.5	70.3	46.3	44.6	34.9	39.8	40.3		
	Ours	78.5	86.0	74.1	17.6	46.0	66.9	36.7	33.5	47.6	29.7	80.8	63.2	81.6	80.7	56.3	49.6	41.5	48.2	48.0		

图 16. 实例分割结果

图 16 中展示了实例分割的结果。和语义分割结果类似，PartSLIP++ 在实例分割也取得了最佳性能，在已发布的 PartSLIP 检查点的基础上，mAP 有显著提高 7.7(48.0 vs. 40.3)。此外，在评估 PartSLIP++ 在 PartNet-E 和 PartNet 之间的 17 个重叠类别时，即使 PartSLIP++ 只在每个类别中训练了 8 个形状，它的性能也超过了在 PartNet 数据集的额外 28k 个形状上训练的最佳模型(SoftGroup)，mAP 提高了 5.2(47.6 vs. 42.4)。这些结果表明 PartSLIP++ 是小样本 3D 实例分割的强大模型。

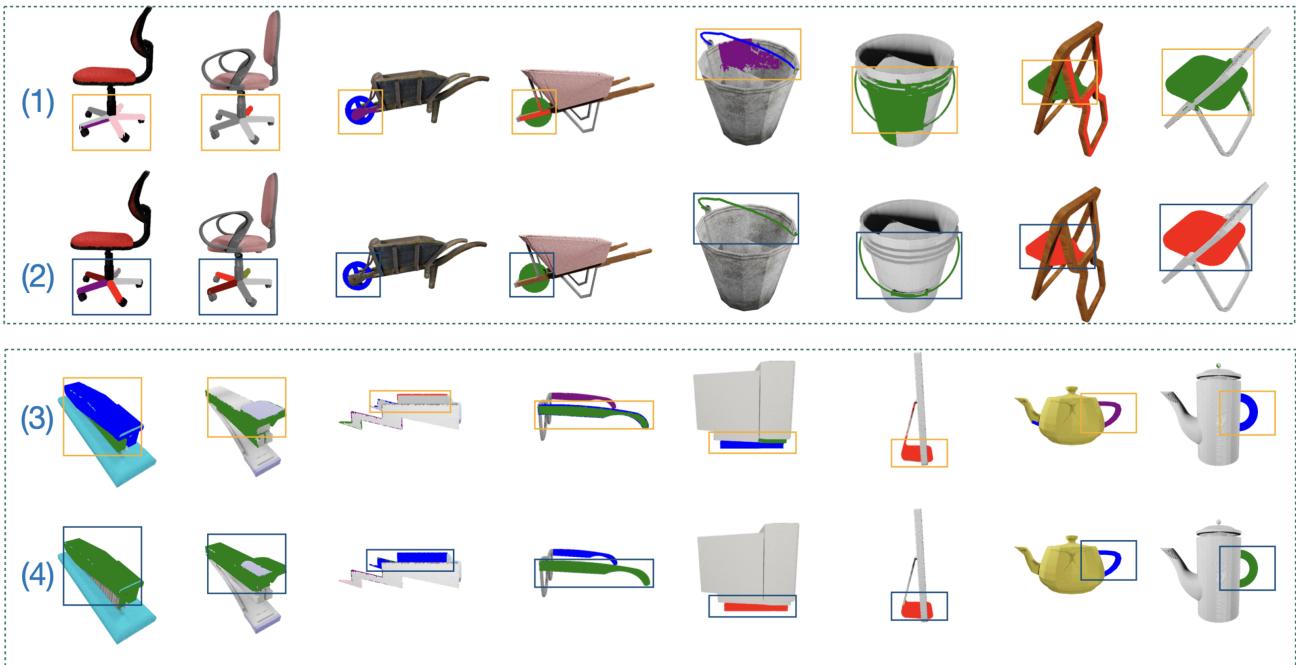


图 17. 实例分割结果可视化

图 17 中可视化比较了 PartSLIP++ 和 PartSLIP 之间的 3D 实例分割质量，其中(1)(3)是 PartSLIP 的结果，(2)(4)是 PartSLIP++ 的结果。可视化结果表明，PartSLIP++ 在生成更精确、噪音更少的 3D 实例分割方面更优越。值得注意的是，在挑战性任务中，比如分割细长的水桶柄、电脑显示器底座或折叠椅的座位时，PartSLIP++ 展现出更高的准确性。与 PartSLIP 生成的分割经常延伸到目标区域外不同，PartSLIP++ 生成的分割保持了更高的精确性和对正确对象部分的粘附性。

参考文献

- [1] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.
- [2] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4558–4567, 2018.
- [3] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975, 2022.
- [4] Minghua Liu, Yinhao Zhu, Hong Cai, Shizhong Han, Zhan Ling, Fatih Porikli, and Hao Su. Partslip: Low-shot part segmentation for 3d point clouds via pretrained image-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21736–21746, 2023.

- [5] Yuchen Zhou, Jiayuan Gu, Xuanlin Li, Minghua Liu, Yunhao Fang, and Hao Su. Partslip++: Enhancing low-shot 3d part segmentation via multi-view instance segmentation and maximum likelihood estimation. *arXiv preprint arXiv:2312.03015*, 2023.